AD-A133 104

Research Note 82-27

TRAINING EFFECTIVENESS AS A FUNCTION OF TRAINING DEVICE FIDELITY: APPENDIXES

David R. Baum and Sharon L. Riedel
Honeywell Systems and Research Center

INSTRUCTIONAL TECHNOLOGY SYSTEMS TECHNICAL AREA

DTIC ELECTE SEP 3 0 1983 S D A

U. S. Army

Research Institute for the Behavioral and Social Sciences

August 1982

83 09 29 034

DTIC FILE COPY

Approved for public release; distribution unlimited.

This report, as submitted by the contractor, has been cleared for release to Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or other reference services such as the National Technical Information Service (NTIS). The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Research Note 82-27
2. GOV'T ACCESSION NUMBER: AD-A133104
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (AND SUBTITLE): TRAINING EFFECTIVENESS AS A FUNCTION OF TRAINING DEVICE FIDELITY: APPENDIXES
5. TYPE OF REPORT/PERIOD COVERED: Final Report March 1981 to August 1982
6. PERFORMING ORG. REPORT NUMBER: 82SRC37

7. AUTHOR(S): David R. Baum, Sharon L. Riedel
8. CONTRACT OR GRANT NUMBER(S): MDA 903-81-C-0214

9. PERFORMING ORGANIZATIONS NAME/ADDRESS: Honeywell Systems and Research Center, 2600 Ridgway Parkway, PO Box 312, Minneapolis, Minnesota 55440
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 20162717A790

11. CONTROLLING OFFICE NAME/ADDRESS: US Army Research Institute for the Behavioral and Social Sciences (PERI-IC), 5001 Eisenhower Avenue, Alexandria, Virginia 22333
12. REPORT DATE: August 1982
13. NUMBER OF PAGES: 118

14. MONITORING AGENCY NAME/ADDRESS (IF DIFFERENT FROM CONT. OFF.):
15. SECURITY CLASSIFICATION (OF THIS REPORT): Unclassified
15a. DECLASSIFICATION DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (OF THIS REPORT)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (OF THE ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)

Simulation training	Fidelity specification	Perceptual-motor skill
Simulators	Fidelity requirements	Maintenance training
Training devices	Fidelity research	
Fidelity	Training device effectiveness	

20. ABSTRACT (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)

This volume contains appendixes related to "Training Effectiveness as a Function of Training Device Fidelity." The appendixes consist of computer program listings and user documentation, raw rating and performance data, illustrations of training devices, and instructions to subjects.

HQ168 REV 11/74

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 55 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

CONTENTS

Accession For
GRA&I
T[illegible]
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1
BEING SCANNED INSPECTED 1

APPENDIX A

SIMTRAIN PILOT PROJECT
USERS MANUAL

INTRODUCTION

THIS DOCUMENT IS A USERS GUIDE FOR THE SIMTRAIN PILOT PROJECT DATA GATHERING SOFTWARE. THE PROGRAM WAS DEVELOPED TO MEASURE BICYCLE WHEEL VARIANCES AT DESIRED INTERVALS DURING THE TRUEING PROCESS.

PREPARATION

THE ELECTRONICS ATTACHED TO THE TRUEING STAND AND THE SOFTWARE TO MONITOR AND RECORD THE MEASUREMENTS ARE EXECUTED ON THE LEFT HAND ECLIPSE COMPUTER IN THE MAN-MACHINE SCIENCES COMPUTER LAB. FOLLOW THE SYSTEM STARTUP PROCEDURES THAT ARE HANGING ON THE RIGHT SIDE OF THE COMPUTER. THE REMOVABLE DISK PACK THAT HAS THE PROGRAMS IS DISK PACK 42. IF THE RESULTS OF THE TRIALS ARE TO BE PRINTED, TURN ON THE PRINTER POWER SWITCH LOCATED ON THE PRINTER PEDESTAL. THE PRINTER FOR THE LEFT ECLIPSE IS THE FLOOR PRINTER NEXT TO THE DOOR.

AFTER COMPLETION OF THE SYSTEM STARTUP PROCEDURE, THE CONSOLE SHOULD HAVE OUTPUT

```
FILENAME?
```

SIMPLY TYPE A RETURN AND ENTER THE DATE AND TIME INFORMATION IN THE REQUESTED FORMAT. THE SYSTEM SHOULD RESPOND WITH AN

```
R
```

WHICH IS THE STANDARD READY PROMPT AND INDICATES THAT THE SYSTEM IS WAITING FOR A USER COMMAND.

DATA GATHERING (WHEELRITE)

THE NAME OF THE DATA GATHERING PROGRAM IS WHEELRITE. ITS FUNCTION IS TO MEASURE THE RIMS'S VARIANCES AT USER DEFINED INTERVALS AND SAVE THE DATA GENERATED IN A UNIQUE FILE WHOSE NAME IS DERIVED BY THE TYPE OF TRAINING GIVEN THE SUBJECT, THE SUBJECT'S NUMBER, AND THE TYPE OF THE EXPERIMENT BEING PERFORMED. THE DATA GENERATED INCLUDE THE VALUES OF THE LOCAL MAXIMA MEASUREMENTS OF THE RIM AND THE SUM AND AVERAGE OF THEIR ABSOLUTE VALUES FOR ONE REVOLUTION OF THE BICYCLE WHEEL. THE DATA IS STORED IN THE DATA FILE, ONE RECORD PER MEASUREMENT INTERVAL. THE DATA RECORD FILE FORMAT IS DESCRIBED IN ATTACHMENT A.

PROCEDURE

1. FROM DISK PACK 42, TYPE THE COMMAND

   WHEELRITE CR (CR = CARRIAGE RETURN)

2. THE USER WILL THEN SUPPLY THE FOLLOWING INFORMATION TO FORM THE DATA FILE NAME

   A. TRAINING METHOD CODE

      THE TRAINING METHOD WILL BE REQUESTED BY THE PROMPT:

      INPUT TRAINER TYPE (A-Z) :

      THE USER ENTERS THE CODE LETTER AND TYPES A RETURN.

   B. SUBJECT ID

      THE USER WILL BE REQUESTED TO ENTER THE SUBJECT'S ID NUMBER WITH THE PROMPT:

      INPUT SUBJECT ID (1-99):

      THE USER ENTERS AN ID NUMBER AND TYPES A RETURN.

   C. TEST SEQUENCE CODE

      THE USER WILL THEN BE ASKED TO ENTER A TEST SEQUENCE CODE AFTER THE PROMPT:

      INPUT TEST ID (2 CHARACTERS):

THE USER ENTERS ONE OR TWO CHARACTERS TO IDENTIFY THE TEST BEING PERFORMED. FOR EXAMPLE, A POSSIBLE SERIES OF CODES FOR THREE PRACTICE RUNS MIGHT BE P1, P2, AND P3.

THE DATA FILE NAME WILL THEN BE GENERATED USING THESE INPUTS. THE FORMAT OF THIS NAME IS

TSS.CC

WHERE

T IS THE TRAINING CODE LETTER
SS IS THE SUBJECT ID NUMBER. (IF THE NUMBER ENTERED IS LESS THAT 10, THIS PART OF THE NAME WILL BE OF THE FORM 0S), AND
CC IS THE TEST SEQUENCE CODE LETTER OR LETTERS.

3. THE PROGRAM WILL THEN RESPOND WITH THE FOLLOWING LINES:

READY FOR INITIAL MEASUREMENT
TO STOP THE PROGRAM, TYPE THE LETTER S
TO MEASURE WHEEL, START WHEEL TURNING AND STRIKE ANY KEY

THE PROGRAM IS NOW READY TO TAKE THE INITIAL MEASUREMENT. WHEN READY TO DO THE MEASUREMENT, START THE WHEEL TURNING, IN EITHER DIRECTION, WITH SUFFICIENT SPEED TO ENSURE THAT AT LAST TWO REVOLUTIONS WILL OCCUR AFTER STRIKING A KEY ON THE KEYBOARD. WHEN THE MEASUREMENT IS COMPLETED, THE PROGRAM RESPONDS WITH THE OUTPUT

INITIAL MEASUREMENT COMPLETE

THE PROGRAM MAY BE TERMINATED BEFORE THIS MEASUREMENT BY TYPING THE LETTER 'S'.

4. AFTER THIS INITIAL MEASUREMENT, THE PROGRAM WILL OUTPUT THE MESSAGE

READY FOR MEASUREMENT N
TO STOP PROGRAM, TYPE THE LETTER S
TO MEASURE WHEEL, START WHEEL TRUNING AND STRIKE ANY KEY

THE PROGRAM IS NOW READY TO COLLECT THE DATA UPON COMMAND. IF THE DATA COLLECTION PROCESS IS COMPLETE, STRIKE THE 'S' KEY. THE PROGRAM RESPONDS WITH THE MESSAGE:

MEASUREMENT PROCESS COMPLETE
NORMAL TERMINATION
DATA WRITTEN TO FILE TSS.CC

WHERE TSS.CC IS THE NAME OF THE DATA FILE DERIVED FROM THE INPUTS OF STEP 2.

TO TAKE ANOTHER MEASUREMENT, START THE WHEEL TURNING, IN EITHER DIRECTION, AT SUFFICIENT SPEED, AND STRIKE ANY KEY (EXCEPT S) ON THE KEYBOARD. THE PROGRAM WILL RESPOND WITH THE MESSAGE

MEASUREMENT N COMPLETE

WHEN THE MEASUREMENT PROCESS IS COMPLETED. THE VALUE OF N IN THIS MESSAGE, AND IN THE READY MESSAGE, IS THE CURRENT SEQUENCE NUMBER IN THE SERIES OF MEASUREMENTS.

POSSIBLE ERRORS DURING THE PROCEDURE

1. TO CORRECT A TYPING MISTAKE DURING THE DATA ENTRY IN STEP 2, SIMPLY TYPE THE '\' KEY ON THE KEYBOARD AND RETYPE THE DATA. THIS MAY BE DONE AS MANY TIMES AS NECESSARY PRIOR TO USING THE RETURN KEY.

2. IF THE VALUES ENTERED DURING STEP 2 GENERATE A FILE NAME OF A FILE THAT ALREADY EXISTS ON THE DISK, THE MESSAGE

FILE TSS.CC ALREADY EXISTS
DO YOU WISH TO OVERWRITE (Y OR N) ?

APPEARS.

THE VALUE TSS.CC WILL BE REPLACED BY THE FILENAME GENERATED. IF THE QUESTION IS ANSWERED WITH THE LETTER 'Y', THE DATA FILE OF THAT NAME ON THE DISK WILL BE DELETED AND A NEW FILE OF THE SAME NAME WILL BE CREATED. THE DATA OF THE OLD FILE WILL BE LOST. ANSWERING THE QUESTION WITH 'N' (OR AND OTHER LETTER) WILL CAUSE THE PROGRAM TO REQUEST REENTRY OF THE INFORMATION FOR STEP 2.

3. TYPING THE LETTER 'S' AFTER THE READY MESSAGE IN STEP 3 AUTOMATICALLY DELETES THE DATA FILE (WHICH WILL BE EMPTY) FROM THE DISK. THE MESSAGE

NO DATA RECORDED. DATA FILE TSS.CC DELETED FROM DISK

WHERE TSS.CC IS THE FILE NAME GENERATED BY STEP 2.

DATA EXAMINATION (EXAMINE)

AFTER THE DATA HAS BEEN GENERATED AND STORED IN THE FILE, THIS PROGRAM IS USED TO DISPLAY IT IN A HUMAN READABLE FORM.

PROCEDURE

1. FROM DISK PACK 42, TYPE THE COMMAND

   EXAMINE CR (CR = CARRIAGE RETURN)

2. SUPPLY THE SAME INFORMATION TO FORM THE DATA FILE NAME AS WAS DONE IN STEP 2 OF THE DATA COLLECTION PROCEDURE. THE ONLY DIFFERENCE IS IF THE VALUES ENTERED GENERATE A FILE NAME OF A NON-EXISTENT FILE, AN ERROR MESSAGE WILL BE OUTPUT AND A REQUEST FOR A NEW SET OF INFORMATION WILL BE ISSUED. TO ABORT THE PROGRAM IN THIS STEP, STRIKE THE CTRL AND 'A' KEYS SIMULTANEOUSLY.

3. WHEN A DATA FILE HAS BEEN DEFINED AND OPENED, A QUESTION IS ASKED WHETHER THE OUTPUT SHOULD GO TO THE CRT OR THE PRINTER. THIS IS DONE WITH THE MESSAGE:

   OUTPUT TO PRINTER ? (0=NO, 1=YES)

4. THE USER IS THEN ASKED WHICH RECORD OF DATA HE WOULD LIKE TO SEE (OR PRINT). THE LEGAL OPTIONS ARE:

-1	STOP THE PROGRAM
0	OUTPUT ALL DATA RECORDS AT ONCE
1 TO N	OUTPUT SPECIFIED DATA RECORD

   IF THE SPECIFIED DATA RECORD DOES NOT EXIST, THE REQUEST MUST BE REENTERED. THE PROGRAM WILL CONTINUE TO CYCLE IN THIS STEP UNTIL A -1 IS ENTERED.

SYSTEM SHUTDOWN

THIS SECTION DESCRIBES THE MEANS TO SHUT DOWN THE SYSTEM IN AN ORDERLY FASHION.

WHEN THE SYSTEM IS IN THE COMMAND PROCESSING MODE, AS DESCRIBED IN THE PREPARATION SECTION, SIMPLY TYPE THE COMMAND

END CR (CR = CARRIAGE RETURN)

THE SYSTEM WILL RESPOND WITH MESSAGES THAT CERTAIN DIRECTORIES HAVE BEEN CLEARED AND THAT THE MASTER DEVICE HAS BEEN RELEASED. WHEN THIS RELEASE MESSAGE APPEARS, GO TO THE COMPUTER AND FOLLOW THE SYSTEM SHUTDOWN PROCEDURES STARTING WITH STEP NUMBER 4. IF THE PRINTER WAS USED, TURN THE PRINTER POWER OFF BEFORE USING THE KEY TO TURN THE COMPUTER POWER OFF.

ATTACHMENT A - DATA FILE RECORD FORMAT

FILE STRUCTURE

THERE IS ONE RECORD IN THE DATA FILE FOR EACH MEASUREMENT TAKEN DURING THE EXPERIMENT. THE DATA IS STORED IN BINARY FORMAT AND WAS CREATED USING THE RECORD READ/WRITE ROUTINE OF FORTRAN 5 (READR AND WRITR). THE INITIAL MEASUREMENT IS IN RECORD ONE WITH ALL SUBSEQUENT MEASUREMENTS FOLLOWING SEQUENTIALLY.

RECORD STRUCTURE

1. RECORD LENGHT - 208 WORDS (416 BYTES)

2. COMPOSITION AND ACCESS

THE DATA FILE RECORDS MAY BE ACCESSED BY DECLARING A COMMON BLOCK WITH THE FOLLOWING ELEMENTS.

```
COMMON /RECORD/ ITIME(3),SUM,AVG,N,VALS(100)
```

WHERE

ITIME - TIME OF DATA RECORD WAS WRITTEN TO FILE. (ITIME(1) = HOURS, ITIME(2) = MINUTES, AND ITIME(3) = SECONDS). INTEGER ARRAY
SUM - SUM OF THE ABSOLUTE VALUES OF THE LOCAL MAXIMA. REAL
AVG - AVERAGE OF THE ABSOLUTE VALUES OF THE LOCAL MAXIMA. REAL
N - NUMBER OF LOCAL MAXIMA DETECTED. INTEGER
VALS - ACTUAL LOCAL MAXIMA VALUES DETECTED. REAL ARRAY

THE DATA RECORDS MAY BE ACCESSED USING THE READR I/O ROUTINE OF FORTRAN 5 AFTER OPENING THE DATA FILE WITH A RECORD LENGTH OF 416 BYTES.

APPENDIX B

COMPUTER PROGRAM LISTING
WHEEL MEASUREMENT

```
      SUBROUTINE WHELRITE

C
C           SIMTRAIN PILOT TEST DRIVER
C

      COMMON /RITE/ FREC,FNAME
      INTEGER FREC              !FILE DATA RECORD NUMBER
      REAL %AXVAL(100)          !LOCAL MAXIMA DATA BLOCK
      INTEGER NMAX              !NUMBER OF LOCAL MAXIMA
      INTEGER FNAME(4)

      INCLUDE "PARAMETERS.IF"

C
C           BEGIN
C           INITIALIZE RUN (#INIT);
C           DATA FILE RECORD NUMBER (FREC) = 1;
C

      X     TYPE "<12><11>SIMTRAIN DATA MEASUREMENT PROGRAM"
      X     CALL #INIT(FNAME,FREC)

C
C           DO UNTIL USER WISHES TO STOP PROGRAM
C               TELL USER TO START SAMPLE OR STOP PROGRAM
C               IF USER WISHES TO MEASURE WHEEL THEN
C                   MEASURE VARIANCES AND LOCATE LOCAL MAXIMA FOR ONE
C                       REVOLUTION OF THE WHEEL (#SAMPLE);
C                   TELL USER THAT MEASUREMENTS ARE COMPLETE;
C                   WRITE DATA OUT TO FILE AND RECORD FREC
C                   IF USER VERIFIES THAT DATA IS VALID THEN
C                       CALCULATE STATISTICS AND OUTPUT TO DATA FILE
C                       INCREMENT DATA FILE RECORD NUMBER (FREC);
C                   ENDIF
C               ENDIF
C           ENDDO
C

   10 CONTINUE
      X     IF(FREC .EQ. 1) TYPE "<12>READY FOR INITIAL MEASUREMENT"
            IF(FREC .NE. 1) TYPE "<12>READY FOR MEASUREMENT",FREC-1
      X     TYPE "TO STOP PROGRAM, TYPE THE LETTER S"
            IF (FREC .EQ. 1) GO TO 16
            TYPE "TO MEASURE WHEEL, STRIKE ANY KEY"
   15 CONTINUE
            CALL GCHAR(ICHAR,IER)
      X     IF(ICHAR .EQ. 'S') GO TO 20
      X     IF(ICHAR .NE. 40K) GO TO 15

   16 CONTINUE
            CALL #SAMPLE(NMAX,MAXVAL)

      X     TYPE "NUMBER OF LOCAL MAXIMA = ",NMAX

      X     IF(FREC .EQ. 1) TYPE "<12><HFL>INITIAL MEASUREMENT COMPLETE"
```

```
      IF(FREC .NE. 1) TYPE "<12><REL>MEASUREMENT",FREC-1," COMPLETE"

      IF (FREC .NE. 1) CALL MAXPRINT(NMAX,MAXVAL)
      CALL WSTATS(FREC,NMAX,MAXVAL)
      CALL UPDATE(FCHAN,IER)
      IF (FREC .EQ. 1) GO TO 17
      WRITE(10,400)
  400 FORMAT(/"DO YOU WANT TO RECORD THIS SET OF MAXIMA ? (Y OR N) ",Z)
      CALL GCHAR(ICHAR,IER)
      CALL PCHAR(ICHAR,IER)
      TYPE " "
      IF(ICHAR .NE. LETY) GO TO 20

   17 CONTINUE
      FREC = FREC + 1
X     GO TO 10

   20 CONTINUE

C
C           CLOSE DATA FILE
C           IF NO DATA RECORDS WERE RECORDED THEN
C               DELETE FILE FROM DISK
C           ENDIF
C           END
C

X     CLOSE FCHAN

X     TYPE "<15><12>MEASUREMENT PROCESS COMPLETE"
X     TYPE "NORMAL TERMINATION"
X     IF (FREC .EQ. 1) GO TO 30
X         WRITE(10,500) FNAME(1)
X 500     FORMAT("DATA WRITTEN TO FILE ",S6)
X         GO TO 40
X  30 CONTINUE
X         WRITE(10,510)FNAME(1)
X 510     FORMAT("NO DATA RECORDED.  DATA FILE ",S6," DELETED FROM DISK")
X         CALL FDELETE (FNAME)
X  40 CONTINUE
X

          RETURN
          END
```

```
      C
      C              DATA FILE INITIALIZATION FOR WHEELRITE PROGRAM
      C
                     SUBROUTINE WINIT(FNAME,FREC)

                     INTEGER FNAME(4)                ! FILE NAME STRING
                     INTEGER FREC                    ! OUTPUT FILE RECORD POINTER
                     INTEGER TRAINER                 ! TRAINER TYPE
                     INTEGER SUBJECT                 ! SUBJECT ID
                     INTEGER TEST                    ! TEST ID
                     INTEGER ISTAT(10)               ! LOCAL TEMPORARY
                     LOGICAL RESTART                 ! EXPERIMENT RESTART FLAG

                     INCLUDE "PARAMETERS.IF"

      C
      C              BEGIN
      C              ENABLE A/D CONVERTER AND DISCRETE I/O DEVICES
      C              SET FILE RECORD NUMBER TO 1  /* INITIAL RECORD */
      C              IF USER WANTS TO RESTART AN EXPERIMENT THEN
      C                  ASK USER TO SPECIFY WHERE TO CONTINUE
      C              ENDIF
      C              DO WHILE FILE DOES NOT EXIST OR USER WISHES TO OVERWRITE
      C                  PROMPT USER FOR TRAINER TYPE;
      C                  PROMPT USER FOR SUBJECT NUMBER;
      C                  PROMPT USER FOR TEST ID;
      C                  BUILD FILE NAME
      C                  IF THIS IS A RESTART THEN EXIT DO LOOP ENDIF
      C                  IF FILE ALREADY EXISTS THEN
      C                      ASK USER IF HE WISHES TO OVERWRITE
      C                  ENDIF
      C              ENDDO
      C

      X              CALL DEBL(42K)          ! DISCRETE I/O DEVICE (DIO)
      X              CALL DEBL(21K)          ! A/D CONVERTER (ADCV)

                     FREC = 1
                     RESTART = .FALSE.
      X              WRITE(10,490)
      X       490    FORMAT("<NL>DO YOU WANT TO RESTART AN EXPERIMENT (Y OR N) ?  ",Z)
      X              CALL GCHAR(ICHAR,IER)
      X              CALL PCHAR(ICHAR)
      X              TYPE " "
      X              IF(ICHAR .NE. LETY) GO TO 10
      X                  RESTART = .TRUE.
      X              ACCEPT "WHAT IS THE NUMBER OF THE NEST MEASURMENT TO RECORD ?  ",FREC
      X              FREC = FREC + 1
          10         CONTINUE
                     ACCEPT "INPUT EXERCISE NUMBER (1-3) : ",IEX
                     CALL INWHEEL(IEX)
                     WRITE(10,500)
            500      FORMAT(/"INPUT TRAINER TYPE (A-Z): ",Z)
                     READ (11,510) TRAINER
            510      FORMAT(S1)
```

```
57
58              ACCEPT "INPUT SUBJECT ID (1-99): ",SUBJECT
59              SUBJECT = MOD(SUBJECT,100)
60
61              WRITE(10,520)
62        520   FORMAT("INPUT TEST ID (2 CHARACTERS): ",Z)
63              READ(11,530) TEST
64        530   FORMAT(S2)
65
66              ENCODE (FNAME,540) TRAINER,TEST
67        540   FORMAT(S1,2X,".",S2,"<0>")
68              BYTE(FNAME,2) = SUBJECT/10 + 60K
69              BYTE(FNAME,3) = MOD(SUBJECT,10) + 60K
70              IF(RESTART) GO TO 110
71
72              CALL STAT(FNAME,ISTAT,IER)
73              IF(IER .EQ. NOFILE) GO TO 100
74        X     TYPE "IER FROM STAT = ",IER
75              WRITE(10,550) FNAME(1)
76        550   FORMAT(/"<BEL>FILE ",S6," ALREADY EXISTS"/
77           &          "DO YOU WISH TO OVERWRITE ? (Y OR N) ",Z)
78              READ(11,510)IANS
79              IF(BYTE(IANS,1) .EQ. LETY) GO TO 100
80              GO TO 10
81
82        100   CONTINUE
83
84        C
85        C               IF NOT A RESTART THEN
86        C                   DELETE ANY EXISTING FILE WITH SAME NAME
87        C               ENDIF
88        C               OPEN DATA FILE
89        C               END
90        C
91
92              DELETE FNAME
93        110   CONTINUE
94              WRITE(10,999) FNAME(1)
95        999   FORMAT ("DATA FILE NAME IS ",S6)
96              OPEN FCHAN,FNAME,LEN = FLEN
97              RETURN
98              END
```

```
      C
      C              WHEEL VARIANCE MEASUREMENT AND LOCAL MAXIMA RECORDING ROUTINE
      C
      
            SUBROUTINE WSAMPLE(NMAX,MAXVAL)

            INTEGER NMAX                        ! NUMBER OF LOCAL MAXIMA
            REAL MAXVAL(100)                    ! LOCAL MAXIMA VALUES

            REAL WDATA(38)
            INTEGER WINDX
            INTEGER REVSTART

      C
      C              BEGIN
      C              GET THE RAW WHEEL DATA FOR ONE REVOLUTION
      C              FIND ALL LOCAL MAXIMA IN DATA
      C              FILTER OUT NOISE POINTS IN DATA
      C              END
      C

            CALL GETDATA(WDATA,WINDX,REVSTART)
            CALL FINDMAX(WDATA,WINDX,REVSTART,MAXVAL,NMAX)
            CALL FILTERMAX(MAXVAL,NMAX)

            RETURN
            END
```

```
      1              SUHROUTINE GETDATA(WDATA,WINDX,WEVSTART)
      2
      3              COMMON /MOD/ YS(36)
      4
      5              REAL WDATA(1)
      6              INTEGER WINDX,WEVSTART
      7
      8              DO 10 I = 1,36
      9                 WDATA(I) = YS(I) * 2.
     10
     11     10       CONTINUE
     12              WDATA(37) = WDATA(1)
     13              WDATA(38) = WDATA(2)
     14              WINDX = 38
     15              WEVSTART = 3
     16
     17              RETURN
     18              END
```

```
      C
      C       MAXIMA DETERMINATION ROUTINE
      C
C
      SUBROUTINE FINDMAX(WDATA,WINDX,REVSTART,MAXVAL,NMAX)

      REAL MAXVAL(1)
      REAL WDATA(1)
      INTEGER WINDX
      INTEGER REVSTART

      REAL DIFF
      INTEGER DIRECTION
      INTEGER UP
      INTEGER DOWN
      REAL NVAL
      REAL OVAL
      REAL CMAX

      INCLUDE "PARAMETERS.IF"

      C
      C       LOGICAL FUNCTION SAMESIGN (TRUE IF I,J ARE SAME ARITHMETIC SIGN)
      C

      SAMESIGN(I,J) = (I .LE. 0 .AND. J .LT. 0) .OR. (I .GT. 0 .AND. J .GE. 0)

      C
      C               BEGIN
      C
      C               /* DETERMINE INITIAL DIRECTION OF DATA */
      C
      C               FIND DIFFERENCE OF FIRST TWO ELEMENTS OF DATA
      C               IF DIFFERENCE IS POSITIVE OR ZERO THEN
      C                   DIRECTION IS UP
      C                   SET CMAX TO SECOND DATA ELEMENT
      C               ELSE
      C                   DIRECTION IS DOWN
      C                   SET CMAX TO FIRST DATA ELEMENT
      C               ENDIF
      C
      C               /* INITIALIZE VARIABLES */
      C
      C               NMAX = 0; OVAL = SECOND DATA ELEMENT;
      C

      UP = 1
      DOWN = -1
      DIFF = WDATA(2) - WDATA(1)
      IF(DIFF .LT. 0) GO TO 10
          DIRECTION = UP
          CMAX = WDATA(2)
          GO TO 20
   10 CONTINUE
          DIRECTION = DOWN
          CMAX = WDATA(1)
```

```
      20    CONTINUE

            NMAX = 0.0
            OVAL = WDATA(2)

C
C               FOR I = 3 TO WINDX DO
C                   NVAL = WDATA(I)
C                   CALCULATE DIFFERENCE OF ADJACENT VALUES NVAL AND OVAL
C
C                   IF DIRECTION IS UP THEN
C                       IF DIFFERENCE IS GREATER THAN ZERO THEN
C                           IF NVAL IS GREATER THAN CMAX THEN
C                               CMAX = NVAL
C                           ENDIF
C                       ELSE
C                           IF CMAX - NVAL IS GREATER THAN NOISE THEN
C                               IF I IS >= REVOLUTION START POINTER AND
C                                  CMAX IS SAME SIGN AS DIRECTION THEN
C                                       INCREMENT NMAX
C                                       PUT SCALED VALUE OF CMAX IN MAXVAL(NMAX)
C                               ENDIF
C                               CHANGE DIRECTION
C                           ENDIF
C                       ENDIF
C

            DO 90 I = 3,WINDX

            NVAL = WDATA(I)
            DIFF = NVAL - OVAL

            IF(DIRECTION .EQ. DOWN) GO TO 50
                IF(DIFF .LE. 0) GO TO 30
                    IF(NVAL .GT. CMAX) CMAX = NVAL
                    GO TO 80
      30        CONTINUE
                    IF(CMAX-NVAL .LE. XNOISE) GO TO 80
                        IF( I .LT. REVSTART .OR.
     &                     .NOT. SAMESIGN(CMAX,DIRECTION)) GO TO 40
                         NMAX = NMAX + 1
                         MAXVAL(NMAX) = FLOAT(CMAX)
      40                CONTINUE
                        DIRECTION = DOWN
                        GO TO 80
      50    CONTINUE

C
C                       ELSE        /* DIRECTION IS DOWN */
C                           IF DIFFERENCE IS LESS THAN ZERO THEN
C                               IF NEW VALUE IS LESS THAN CMAX THEN
C                                   CMAX = NVAL
C                               ENDIF
C                           ELSE
C                               IF NVAL-CMAX IS GREATER THAN NOISE THEN
C                                   IF I IS PAST START OF REVOLUTION POINTER AND
```

```
C                                   CMAX IS SAME SIGN AS DIRECTION THEN
C                                    INCREMENT NMAX
C                                    PUT SCALE VALUE OF CMAX IN MAXVAL
C                                 ENDIF
C                                 CHANGE DIRECTION
C                              ENDIF
C                           ENDIF
C                        ENDIF
C                        SET OVAL TO NVAL
C                     ENDDO
C
C                     END
C
              IF(DIFF.GE. 0) GO TO 60
                  IF(NVAL .LT. CMAX) CMAX = NVAL
                  GO TO 80
          60  CONTINUE
                  IF(NVAL-CMAX .LE. XNOISE) GO TO 80
                      IF(I .LT. REVSTART .OR.
            &            .NOT. SAMESIGN(CMAX,DIRECTION)) GO TO 70
                           NMAX = NMAX + 1
                           MAXVAL(NMAX) = FLOAT(CMAX)
          70          CONTINUE
                      DIRECTION = UP
                      GO TO 80

          80      CONTINUE
                  OVAL = NVAL

          90  CONTINUE

      X       WRITE(12,500)NMAX,(K,(MAXVAL(I),I=K,K+4),K=1,NMAX,5)
      X 500   FORMAT("1"//"   DUMP FROM FINDMAX"//" NUMBER OF MAX = ",I4/
      X     &          /20(I4,":",5(F17.5)))
              RETURN
              END
```

```
      C
      C       MAXIMA FILTER ROUTINE
      C
C
      SUBROUTINE FILTERMAX(MAXVAL,NMAX)

      REAL MAXVAL(1)
      INTEGER NMAX

      REAL RNOISE
      INTEGER LAST

      INCLUDE "PARAMETERS.IF"

C
C             BEGIN
C
C             INITIALIZE LAST TO 0
C             FOR I = 1 TO NMAX DO
C                 IF MAXVAL(I) IS NOT NOISE THEN
C                     INCREMENT LAST
C                     SET MAXVAL(LAST) TO MAXVAL(I)
C                 ENDIF
C             ENDDO
C             SET NMAX TO LAST
C
C             END
C

      LAST = 0
      RNOISE = XNOISE
X     RNOISE = FLOAT(NOISE) * SCALE
      DO 10 I = 1,NMAX
          IF(ABS(MAXVAL(I)) .LE. RNOISE) GO TO 10
              LAST = LAST + 1
              MAXVAL(LAST) = MAXVAL(I)
   10 CONTINUE

      NMAX = LAST

X     WRITE(12,500)NMAX
X500  FORMAT("1"//"   DEBUG DUMP FROM FILTERMAX"//" NUMBER OF MAXIMA = ",I4)
X     WRITE(12,510)(K,(MAXVAL(I),I=K,K+4),K=1,NMAX,5)
X510  FORMAT(I4,":",5(F17.5))

      RETURN
      END
```

```
      C
      C                LOCAL MAXIMA CONSOLE OUTPUT ROUTINE
      C

               SUBROUTINE MAXPRINT(NMAX,MAXVAL)

               INTEGER NMAX
               REAL MAXVAL(NMAX)

               WRITE(10,500) NMAX
          500  FORMAT(/"THE NUMBER OF LOCAL MAXIMA FOUND WAS ",I3/
              &          "THE VALUES OF THE LOCAL MAXIMA ARE:")

               DO 10 I = 1,NMAX,5
               J = I + 4
               IF(J .GT. NMAX) J = NMAX
               WRITE(10,510) I,(MAXVAL(K),K=I,J)
          510  FORMAT(I3,":",1X,5(F14.9))
           10  CONTINUE

               RETURN
               END
```

```
C
C                MAXIMA STATISTICS AND RECORDING ROUTINE
C

                SUBROUTINE WSTATS(FREC,NMAX,MAXVAL)

                INTEGER FREC                    ! DATA FILE RECORD NUMBER
                INTEGER NMAX                    ! NUMBER OF LOCAL MAXIMA DETECTED
                REAL MAXVAL(NMAX)               ! LOCAL MAXIMA VALUES

                COMMON /DATAREC/ ITIME,SUM,AVG,N,VALS
                INTEGER ITIME(3)                ! TIME OF DAY DATA RECORDED (H,M,S)
                REAL SUM                        ! SUM OF MAXIMA VALUES
                REAL AVG                        ! AVERAGE OF MAXIMA VALUES
                INTEGER N                       ! NUMBER OF MAXIMA VALUES
                REAL VALS(100)                  ! MAXIMA VALUES

                INCLUDE "PARAMETERS.IF"

C
C                BEGIN
C                SUM MAXIMA VALUES;
C                AVERAGE MAXIMA VALUES;
C                OUTPUT SUM, AVERAGE, NUMBER OF MAXIMA, AND VALUES TO FILE
C                END
C

                SUM = 0.0
                N = NMAX

                DO 10 I = 1,NMAX
                VALS(I) = MAXVAL(I)
                SUM = SUM + ABS(VALS(I))
             10 CONTINUE

                AVG = SUM / FLOAT(NMAX)

             X  TYPE "AVG,SUM,NMAX :",AVG,SUM,NMAX

                CALL TIME(ITIMF,IER)

                CALL WRITW(FCHAN,FREC,ITIMF,1,IER)
                IF(IER .NE. 1) TYPE "WRITW ERROR (WSTATS).  FREC,IER :",FREC,IER

                RETURN
                END
```

```
      SUBROUTINE CFCHAN

C---
C---  ROUTINE TO CLOSE DATA FILE
C---
      COMMON /RITE/ FREC,FNAME

      INTEGER FREC
      INTEGER FNAME(4)

      INCLUDE "PARAMETERS.IF"

      CLOSE FCHAN
      TYPE "<NL>PROCESS COMPLETE"
      TYPE "NORMAL TERMINATION"
      IF (FREC .EQ. 1) GO TO 30
      WRITE(10,500) FNAME(1)
500   FORMAT("DATA WRITTEN TO FILE ",36)
      GO TO 40

30    CONTINUE
      WRITE(10,510) FNAME(1)
510   FORMAT("NO DATA RECORDED.  DATA FILE ",36," DELETED FROM DISK")
      CALL FDELETE(FNAME)

40    CONTINUE
      RETURN
      END
```

APPENDIX C

USER DOCUMENTATION
COMPUTER GRAPHICS SIMULATOR

1.0 INTRODUCTION

LEXISIM is an interactive computer graphics simulator developed for training subjects to true a bicycle wheel. The Simulator has low-physical and high-functional similarities to an actual trueing system. The LEXISIM program, developed in FORTRAN on a Lexidata System 3400 Video Image and Graphics Processor communicating with a Data General Eclipse S/200 computer, performs the following tasks:

1. Displays the demonstation graphics the instructor uses to explain the trueing procedures to the subject.

2. Provides the user with the following options for truing the wheel:

    a. Adjust the spokes.
    b. Adjust the calipers.
    c. Turn the wheel.
    d. Stop the wheel.
    e. Change the direction the wheel is turning.
    f. Change the speed of the wheel.

3. Takes periodic measurements of the bicycle wheel variances.

2.0 EXECUTING LEXISIM

2.1 Setup

1. The software for LEXISIM is resident on disk pack #28 and must be executed on the left-hand Eclipse computer in the Man-Machine Sciences lab.

2. Follow the startup procedures next to the computer, loading disk pack #28 in disk drive #0. If trials are also to be run under condition "A" (high physical, high-functional system), load disk pack #42 in disk drive #1.

3. Be sure the red light on the front panel of the Lexidata processor is pushed in. There are two Lexidata processors located above the two disk drives next to the window. The LEXISIM simulator uses the lower one.

4. If trial results are to be printed, turn on the power switch on the printer located next to the door and press the ON/LINE button.

5. Turn on the power switch on the Lexidata monitor. The switch is located on the lower right corner on the front of the monitor. A cross-hair should be displayed on the monitor along with the words "IDOS REV 1.6".

2.2 Program Startup

1. After system startup the system responds with:

   FILENAME?

   Press the carriage return and enter the date and time in the requested format. The system responds with an R to indicate that the system is ready to accept user commands.

2. To start the program, type:

   LEXISIM carriage return

   The system now begins to issue prompts for user inputs.

2.3 Prompts and User Responses

1. The user will first be asked to supply information needed to create a data file for storing wheel measurements. Refer to the "SIMTRAIN PILOT PROJECT DATA COLLECTION" document for appropriate responses to the prompts. In addition, the following prompt will be issued:

   ENTER EXERCISE NUMBER (1-3):

   a. Exercise 1 is a test trial for the instructor to demonstrate the simulator to the subject. No wheel measurements will be taken for this exercise.

   b. Exercises 2 and 3 correspond to practice sessions 1 and 2, respectively, for the trial subject.

2. After supplying the information for the data file, the following menu will be issued:

   1. EQUIPMENT COMPONENTS
   2. FINDING WOBBLE
   3. SPOKE ADJUSTMENT
   4. FINE TUNING

   TYPE 1, 2, 3, 4 (0 TO STOP DEMOS):

   User Responses:

   1. "1" - "4" - the corresponding instructor demonstration will be displayed.

   2. "0" - the simulation will begin.

3. At the start of the simulation the following menu will be issued:

   A : ADJUST SPOKES
   C : ADJUST CALIPERS
   D : CHANGE DIRECTION OF WHEEL MOVEMENT
   S : STOP WHEEL
   T : TURN WHEEL
   W : CHANGE SPEED OF WHEEL

   TYPE A, C, D, S, T, or W:

User Responses:

1. "A" - the following prompt will be issued:

WHICH SPOKE DO YOU WANT TO ADJUST?

TYPE 1, 2, 3,... or 36 (S to STOP SPOKE ADJUSTMENT):

User Responses:

a. "1" - "36" - the following menu will be issued:

1 : TURN SPOKE CLOCKWISE
2 : TURN SPOKE COUNTER-CLOCKWISE
3 : STOP ADJUSTMENTS ON SPOKE n

TYPE 1, 2, or S:

User Responses:

i. "1" or "2" - the spoke will be adjusted in the corresponding direction.
ii. "S" - adjustment is stopped on the current spoke.

b. "S" - adjustment is stopped on all spokes.

2. "C" - the following menu will be issued:

I : MOVE CALIPERS IN
O : MOVE CALIPERS OUT
S : STOP CALIPER ADJUSTMENT

TYPE I, O, OR S:

User Responses:

a. "I" - the calipers will be moved in.
b. "O" - the calipers will be moved out.
c. "S" - caliper adjustment will be stopped.

3. "D" - the wheel will move in the opposite direction it is currently moving.

4. "S" - the wheel will stop turning.

5. "T" - the wheel will start turning.

6. "W" - the following prompt will be issued:

TYPE 1, 2, OR 3 (1 = SLOWEST SPEED, 3 = FASTEST SPEED):

User Responses:

a. "1" - the wheel will move at the slowest speed.
b. "2" - the wheel will move at a medium speed.
c. "3" - the wheel will move at the fastest speed.

7. Whenever the prompt "TYPE A, C, D, S, T, OR W" appears, the instructor may respond with the following options:*

a. "M" - a measurement will be taken of the wheel variances. Again, follow the "SIMTRAIN PILOT PROJECT DATA COLLECTION" document for appropriate responses to the prompts.

b. "Q" - the data file for the trial will be closed and the simulation will end.

* These two options were intentionally left off the main user option menu to prevent the subject from taking wheel measurements or ending the simulation.

2.4 Errors

1. If at any time an inappropriate response is given, the prompt or menu will be issued again until a valid response is given.

2. If the spoke adjustment option is selected while the wheel is still turning, the following message will be issued:

YOU MUST STOP THE WHEEL BEFORE ADJUSTING SPOKES

Simply stop the wheel first and then adjust the spokes.

3. If the wheel hits the calipers while moving, the wheel will stop and a bell will be sounded.

4. If the calipers are adjusted too far in or out, a bell will be sounded.

2.5 Data Examination:

After a simulation run, the data may be examined by using the EXAMINE program. The procedures for using EXAMINE may be found in the "SIMTRAIN PILOT PROJECT DATA COLLECTION" document.

3.0 CONDITION "A" TRIALS

If wheel measurements are to be taken for trials run under condition "A" (high-physical, high-functional system), use the following procedures:

1. When the system is in the command processing mode (the ready prompt, R, appears on the terminal) type the following command:

   DIR DP1 carriage return

2. Type the command:

   WHEELRITE carriage return

3. Follow the procedures in the "SIMTRAIN PILOT PROJECT DATA COLLECTION" document for taking wheel measurements and examining the data.

4. To return to the LEXISIM simulator while in the command processing mode, type the command:

   DIR DPØ carriage return (where Ø = zero)

4.0 SYSTEM SHUTDOWN

The system may be shut down at any time when it is in the command processing mode.

1. If trials were run under condition "A", type the command:

   RELEASE DP1 carriage return

2. Type the command:

   END carriage return

   The system will respond with the messages:

   DIRECTORY DPØ CLEARED
   MASTER DEVICE RELEASED

3. Follow the remaining shutdown procedures next to the computer, starting with step #4.

4. Turn off the power switches on the line printer and Lexidata monitor.

APPENDIX D

COMPUTER PROGRAM LISTING
GRAPHICS DISPLAY DEVICE

```
      C---
      C---  MAIN DRIVER ROUTINE FOR LEXIDATA WHEEL SIMULATOR
      C---

            COMMON /RITE/ FREC,FNAME

            INTEGER FREC
            INTEGER FNAME(4)

      C---
      C---  INITIALIZE DATA FILE FOR WHEEL MEASUREMENTS
      C---

            CALL WINIT(FNAME,FREC)

      C---
      C---  TAKE INITIAL WHEEL MEASUREMENT
      C---

            CALL WHELRITE

      C---
      C---  INITIALIZE LEXIDATA DISPLAY
      C---

            CALL INIT

      C---
      C---  DISPLAY INSTRUCTOR DEMOS
      C---

            CALL DEMOS

      C---
      C---  START SIMULATOR
      C---

            TYPE "<NL>PRESS ANY KEY TO START SIMULATOR<NL>"
            CALL GETC(ICHAR)
            CALL WHEEL

      C---
      C---  CLOSE DATA FILE FOR WHEEL MEASUREMENTS
      C---
            CALL CFCHAN

            STOP
            END
```

```
      SUBROUTINE INIT
C---
C---  ROUTINE TO INITIALIZE THE LEXIDATA
C---
      INTEGER BUF1(3000)

C---
C---  OPEN THE DISPLAY
C---
      CALL DSOPN(48,49,2,IERR)
      IF (IERR .NE. 0) GO TO 999

C---
C---  GET IDO334 MAP
C---
      CALL GETMP(BUF1,IERR1)
      IF (IERR1 .NE. 1) GO TO 900

C---
C---  LOAD MICRO-PROGRAM
C---
      CALL DSPLD(BUF1)
      IF (IERR .NE. 0) GO TO 999

C---
C---  CONFIGURE DISPLAY PARAMETERS
C---
      CALL DSCFG(511,511,8)
      IF (IERR .NE. 0) GO TO 999

C---
C---  CLEAR THE DISPLAY
C---
      CALL DSCLR(-1)
      IF (IERR .NE. 0) GO TO 999

C---
C---  SELECT THE HARDWARE CURSOR
C---
      CALL DSCSL(2,0,0)
      IF (IERR .NE. 0) GO TO 999

C---
C---  ERASE THE MATRIX CURSOR
C---
      CALL DSCER
      IF (IERR .NE. 0) GO TO 999

      GO TO 1000

900   TYPE "IDO334 ERROR !!!",IERR1
      STOP

999   IF (IERR .EQ. 34) TYPE "NO ICHS AVAILABLE"
      IF (IERR .EQ. 37) TYPE "DEVICE CODE ALREADY IN USE"
      IF (IERR .EQ. 49) TYPE "TASK ID ALREADY IN USE"
```

```
      IF (IERR .EQ. 65) TYPE "DEVICE TIMEOUT"
      STOP

1000  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WHEEL

C---
C---  ROUTINE TO PROMPT USER FOR OPTION TO CONTROL THE WHEEL
C---


      COMMON /TEMPS/ IDIRECT,ISTEMP,ISTEMP1,ISPEED
      COMMON /STP/ ISTOP

      DIMENSION INPUT(3)

C---
C---  INITIALIZE VARIBLES AND WHEEL ON THE DISPLAY
C---

      CALL INITIAL

C---
C---  DISPLAY MENU
C---

      CALL MENU

C---
C---  PROMPT USER FOR OPTION
C---

1     CONTINUE
      WRITE(10,3000)
3000  FORMAT(1X,'<NL>TYPE A, C, D, S, T, OR W :  ',Z)
      READ(11,4000) ICHAR
4000  FORMAT(1A1)
      ICHAR = ICHAR - 20100K
      IF ((ICHAR .LT. 1) .OR. (ICHAR .GT. 26)) GO TO 1
      GO TO (2,1,4,3,1,1,1,1,1,1,1,1,9,1,1,1,11,
     +  1,6,7,1,1,8,1,1,1), ICHAR

C---
C---  ADJUST SPOKES
C---

2     CONTINUE
      CALL ADJUST
      CALL MENU
      GO TO 1

C---
C---  CHANGE DIRECTION OF WHEEL
C---

3     CONTINUE
      IDIRECT = -IDIRECT
      CALL MENU
      GO TO 1

      END
```

```
      1              SUBROUTINE INITIAL
      2
      3      C---
      4      C---    ROUTINE TO INITIALIZE VARIABLES AND INITIALIZE THE WHEEL
      5      C---    ON THE DISPLAY
      6      C---
      7
      8              COMMON /TEMPS/ IDIRECT,ISTEMP,ISTEMP1,ISPEED
      9              COMMON /STP/ ISTOP
     10              COMMON /PTS/ NPOINTS(36),NDATA(180,3)
     11              COMMON /SPKS/ SPNUM(5),NSTR(36,2)
     12              COMMON /YLOC/ YCOOR(5)
     13              COMMON /ILIR/ IL,IR
     14              COMMON /LRC/ LEFT,RIGHT,CENTER
     15              COMMON /IPT/ IPOINT
     16              COMMON /CAL/ ILEFT,IRIGHT
     17              COMMON /MOD/ YS(36)
     18
     19              DIMENSION INPUT(3),JSTRING(0:9)
     20
     21              INTEGER CENTER,YCOOR,SPNUM,RIGHT
     22
     23              JSTRING(0) = "0"
     24              JSTRING(1) = "1"
     25              JSTRING(2) = "2"
     26              JSTRING(3) = "3"
     27              JSTRING(4) = "4"
     28              JSTRING(5) = "5"
     29              JSTRING(6) = "6"
     30              JSTRING(7) = "7"
     31              JSTRING(8) = "8"
     32              JSTRING(9) = "9"
     33              IPOINT = 15
     34              ISTOP = 0
     35              ISTEMP = 1
     36              ISPEED = 1
     37              IDIRECT = 1
     38
     39      C---
     40      C---    INITIALIZE NUMBER OF POINTS WHEEL IS OFF CENTER AT EACH POINT
     41      C---    ALONG A REVOLUTION
     42      C---
     43
     44              DO 100 I = 1,36
     45                 NPOINTS(I) = YS(I) * 1000. + .5
     46
     47      100     CONTINUE
     48              CALL UPDAT
     49              DO 111 I = 1,180
     50
     51      111     CONTINUE
     52      C---
     53      C---    INITIALIZE CHARACTER STRINGS FOR SPOKE NUMBERS
     54      C---
     55
     56              DO 10 I = 1,9
```

```
57                NSTR(I,1) = " "
58                NSTR(I,2) = JSTRING(I)
59
60      10      CONTINUE
61              DO 20 I = 10,56
62                NSTR(I,1) = JSTRING(I/10)
63                NSTR(I,2) = JSTRING(MOD(I,10))
64
65      20      CONTINUE
66
67      C---
68      C---    DISPLAY STATIONARY WHEEL
69      C---
70
71              CALL PWHEEL
72
73      C---
74      C---    INITIALIZE SPOKE NUMBERS FOR STATIONARY WHEEL
75      C---
76
77              DO 105 I = 1,5
78                SPNUM(I) = 7-I
79
80      105     CONTINUE
81
82      C---
83      C---    DISPLAY SPOKE NUMBERS ON STATIONARY WHEEL
84      C---
85
86              INTENSITY = -1
87              CALL STANUM(INTENSITY)
88
89      C---
90      C---    INITIALIZE SPOKE NUMBERS AND SPOKE LOCATIONS FOR MOVING WHEEL
91      C---
92
93              DO 55 I = 1,5
94                SPNUM(I) = SPNUM(I) - 1
95                YCOOR(I) = 16 + I * 40
96
97      55      CONTINUE
98
99      C---
100     C---    INITIALIZE LOCATION OF WHEEL
101     C---
102
103             LEFT = IL = NDATA(IPOINT,1)
104             RIGHT = IR = NDATA(IPOINT,2)
105             CENTER = NDATA(IPOINT,3)
106
107     C---
108     C---    DISPLAY INITIAL POSITION OF WHEEL
109     C---
110
111             CALL DSVEC(LEFT,56,LEFT,256,-1)
112             CALL DSVEC(RIGHT,56,RIGHT,256,-1)
```

```
C---
C---    DISPLAY INITIAL SPOKES ON WHEEL
C---

        DO 150 I = 1,5
           CALL DSCIR(CENTER,YCOOR(I),2,-1)
           CALL DSSAD(CENTER-9,YCOOR(I)+4,-1,0,1)
           CALL DSTXT(NSTR(SPNUM(I),1))
           CALL DSTXT(NSTR(SPNUM(I),2))

150     CONTINUE

C---
C---    DISPLAY CALIPERS
C---

        ILEFT = 159
        IRIGHT = 353
        INTEN = -1
        CALL CALIP(ILEFT,IRIGHT,INTEN)

C---
C---    START TASK TO SPIN WHEEL
C---

        TASK SPIN,ID=1,PRI=0

        RETURN
        END
```

```
      SUBROUTINE INWHEFL(EXERCISE)

C---
C---  ROUTINE TO INITIALIZE THE WHEEL DEVIATIONS
C---  FOR EXERCISES 1-3
C---
      COMMON /MOD/ YS(36)
      INTEGER EXERCISE,EX1,EX2,EX3

      DIMENSION EX1(36),EX2(36),EX3(36)

      DATA EX1/30,27,22,14,0,-14,-22,-52,-72,-88,-95,-92,-78,-56,-31,-10,7,17,
     +          17,12,3,-13,-33,-50,-63,-68,-64,-52,-37,-21,-12,-2,8,19,27,29/

      DATA EX2/119,122,113,94,71,48,27,12,3,-1,0,6,9,10,5,-6,-23,-42,-59,
     +         -74,-83,-91,-94,-92,-85,-70,-54,-35,-21,-8,1,13,33,57,82,104/

      DATA EX3/77,83,80,74,65,56,44,30,12,0,-14,-30,-44,-62,-77,-89,-94,
     +         -89,-73,-53,-32,-19,-17,-26,-42,-59,-75,-81,-80,-72,-60,
     +         -39,-14,14,40,62/

      GO TO (1,2,3),EXERCISE

1     CONTINUE
      DO 100 I = 1,36
         YS(I) = EX1(I) / 2000.

100   CONTINUE
      GO TO 4

2     CONTINUE
      DO 200 I = 1,36
         YS(I) = EX2(I) / 2000.

200   CONTINUE
      GO TO 4

3     CONTINUE
      DO 300 I = 1,36
         YS(I) = EX3(I) / 2000.

300   CONTINUE

4     CONTINUE
      RETURN
      END
```

```
      C       GETMP           GET MICROPROGRAM INTO BUFFER
      C       9/7/78          RDOS VERSION

              SUBROUTINE GETMP(IBUF,IERR)
              INTEGER IBUF(1)

              CALL OPEN(2,"IDOS34.PL",1,IERR)
              IF (IERR.NE.1) GOTO 990
              CALL RDBLK(2,0,IBUF,8,IERR)
              CALL CLOSE(2)
      990     RETURN

              END
```

```
C---
C---    ADJUST CALIPERS
C---

4       CONTINUE
        CALL ADJCAL
        CALL MENU
        GO TO 1

C---
C---    STOP WHEEL
C---

6       CONTINUE
        ISTOP = 0
        CALL MENU
        GO TO 1

C---
C---    START WHEEL TURNING
C---

7       CONTINUE
        ISTOP = 1
        ISTEMP = 0
        ISTEMP1 = 0
        WAKEUP 2
        WAKEUP 3
        CALL MENU
        GO TO 1

C---
C---    CHANGE SPEED OF WHEEL
C---

8       CONTINUE
        WRITE(10,5000)
5000    FORMAT ('<NL><NL>TYPE 1, 2, OR 3 (1 = SLOWEST SPEED,
     +  3 = FASTEST SPEED) : ',Z)
        READ(11,6000) (INPUT(I), I = 1,3)
6000    FORMAT(3R1)
        CALL CONVERT(INPUT,IS)
        IF ((IS .LT. 1) .OR. (IS .GT. 3)) GO TO 8
        ISPEED = IS
        IF (ISPEED .NE. 3)  GO TO 400
        ISTEMP = 0
        ISTEMP1 = 0

400     CONTINUE
        CALL MENU
        GO TO 1

C---
C---    MEASURE WHEEL VARIANCES
C---
```

```
      9     CONTINUE
            TYPE "<NL>ARE YOU SURE YOU WANT TO TAKE A MEASUREMENT ? (Y = YES,
     +      N = NO)"
            READ(11,8000) NOYES
            IF ((NOYES .NE. " Y") .AND. (NOYES .NE. " N")) GO TO 9
            IF (NOYES .EQ. " Y") CALL WHELRITE
            CALL MENU
            GO TO 1

C---
C---  END SIMULATOR
C---

     11     CONTINUE
            TYPE "<NL><NL>ARE YOU SURE YOU WANT TO QUIT ? (Y = YES, N = NO)"
            WRITE(10,7000)
7000  FORMAT("<NL>TYPE Y OR N : ",7)
            READ(11,8000) NOYES
8000  FORMAT(A1)
            IF (NOYES .EQ. " Y") GO TO 600
            IF (NOYES .NE. " N") GO TO 11
            CALL MENU
            GO TO 1

600   CONTINUE

            RETURN
            END
```

```
      SUBROUTINE SPIN

C---
C---  TASK TO SPIN THE WHEEL ON THE MONITOR
C---

      COMMON /TEMPS/ JDIRECT,ISTEMP,ISTEMP1,ISPEED
      COMMON /SIP/ ISTOP
      COMMON /PTS/ NPOINTS(36),NDATA(180,3)
      COMMON /CAL/ ILEFT,IRIGHT
      COMMON /SPKS/ SPNUM(5),NSTR(36,2)
      COMMON /YLOC/ YCOOR(5)
      COMMON /ILIR/ IL,IR
      COMMON /LRC/ LEFT,RIGHT,CENTER
      COMMON /IPT/ IPOINT
      COMMON /CNT/ J

      INTEGER LEFT,RIGHT,CENTER,SPNUM,YCOOR

C---
C---  WAIT UNTIL USER STARTS WHEEL TURNING
C---

      IDIRECT = JDIRECT
      WAIT 2
      J = 0

200   CONTINUE
      J = J + IDIRECT
      IF (J .GT. 180) J = 1
      IF (J .LT. 1) J = 180
      IF (ISPEED .NE. 1) GO TO 250

C---
C---  SLOW SPEED
C---

      JSLOW = 20000
      IF (ISPEED .EQ. 2) JSLOW = 5000
      DO 230 KK = 1,5
      DO 225 K = 1,JSLOW
225   CONTINUE
230   CONTINUE

250   CONTINUE
      IF ((ISPEED .EQ. 3) .AND. (ISTEMP .NE. 0)) GO TO 300

C---
C---  ERASE CURRENT SPOKE NUMBERS ON WHEEL
C---

         DO 300 I = 1,5
            CALL DSCIR(CENTER,YCOOR(I),2,0)
            CALL DSSAD(CENTER-9,YCOOR(I)+4,0,0,1)
            ITEMP = 6-I+J/5
            IF (ITEMP .GT. 36) ITEMP = ITEMP - 36
```

```
                  CALL DSTXT(NSTR(ITEMP,1))
                  CALL DSTXT(NSTR(ITEMP,2))

300         CONTINUE
            ISTEMP = 1
            DO 400 I = 1,5
               YCOOR(I) = 16 + I * 40 + 8 * (MOD(J,5) )

400         CONTINUE

C---
C---    CHECK IF WHEEL IS HITTING CALIPERS
C---

               IL = LEFT
               IR = RIGHT
               IPOINT = IPOINT + IDIRECT
               IF (IPOINT .GT. 180) IPOINT = 1
               IF (IPOINT .LT. 1) IPOINT = 180
               LEFT = NDATA(IPOINT,1)
               RIGHT = NDATA(IPOINT,2)
               CENTER = NDATA(IPOINT,3)
               IF ((LEFT .GT. ILEFT) .AND. (RIGHT .LT. IRIGHT)) GO TO 425
               IF (LEFT .GT. ILEFT) GO TO 415
               LEFT = ILEFT + 1
               RIGHT = LEFT + 50
               GO TO 420

415         CONTINUE
            RIGHT = IRIGHT - 1
            LEFT = RIGHT - 50

420         CONTINUE
            CENTER = (LEFT + RIGHT) / 2
            TYPE "<BEL>"
            ISTOP = 0

425         CONTINUE

C---
C---    ERASE OLD RIM
C---

               CALL DSVEC(IL,56,IL,256,0)
               CALL DSVEC(IR,56,IR,256,0)

C---
C---    DISPLAY NEW RIM
C---

               CALL DSVEC(LEFT,56,LEFT,256,-1)
               CALL DSVEC(RIGHT,56,RIGHT,256,-1)

               IF ((ISPEED .EQ. 3) .AND. (ISTOP .EQ. 1)) GO TO 500

C---
```

```
113     C---    DISPLAY NEW SPOKE NUMBERS
114     C---
115
116               DO 500 I = 1,5
117                  IF ((YCOOR(I) .LT. 60) .OR. (YCOOR(I) .GT. 250)) GO TO 500
118                  CALL DSCIN (CENTER,YCOOR(I),2,-1)
119                  CALL DSSAO(CENTER-9,YCOOR(I)+4,-1,0,1)
120                  ITEMP = 6 - I + J/5
121                  IF (ITEMP .GT. 36) ITEMP = ITEMP - 36
122                  CALL DSTXT(NSTR(ITEMP,1))
123                  CALL DSTXT(NSTR(ITEMP,2))
124
125     500          CONTINUE
126                  IF (ISTOP .NE. 0) GO TO 525
127                  IF (ISPEED .NE. 3) GO TO 515
128                  IF (MOD(SPNUM(I),2) .NE. 1) GO TO 510
129
130     C---
131     C---    UPDATE SPOKE NUMBERS FOR STATIONARY WHEEL
132     C---
133
134               DO 505 I = 1,5
135                  SPNUM(I) = SPNUM(I) - 1
136                  IF (SPNUM(I) .LT. 1) SPNUM(I) = 36
137
138     505          CONTINUE
139
140     510          CONTINUE
141
142     C---
143     C---    DISPLAY SPOKE NUMBERS ON STATIONARY WHEEL
144     C---
145
146               INTENSITY = -1
147               CALL SIANUM(INTENSITY)
148
149     515          CONTINUE
150
151     C---
152     C---    WAIT IF USER STOPPED WHEEL
153     C---
154
155               WAIT 2
156
157     525       CONTINUE
158               IF ((ISPEED .EQ. 3) .AND. (ISTEMP1 .EQ. 1))  GO TO 625
159               IF (ISPEED .EQ. 3) GO TO 650
160               IF (MOD(J,5) .EQ. 0) GO TO 625
161               ISLOW = 20000
162               IF (ISPEED .EQ. 2) ISLOW = 5000
163               DO 675 JJ = 1,ISLOW
164     675       CONTINUE
165               GO TO 600
166
167     650       CONTINUE
168               ITEMP = 5 + J/5
```

```
169
170     C---
171     C---    ERASE SPOKE NUMBERS ON STATIONARY WHEEL
172     C---
173
174             INTENSITY = 0
175             CALL STANUM(INTENSITY)
176             ISTEMP1 = 1
177
178     625     CONTINUE
179             DO 700 I = 1,5
180                ITEMP = 6 - I + J/5
181                IF (IDIRECT .EQ. -1) ITEMP = ITEMP -1
182                IF (ITEMP .GT. 36) ITEMP = ITEMP - 36
183                SPNUM(I) = ITEMP
184
185     700     CONTINUE
186             IF ((MOD(SPNUM(1),2) .EQ. 1) .OR. (ISPEED .EQ. 3)) GO TO 600
187
188     C---
189     C---    DISPLAY NEW SPOKE NUMBERS ON STATIONARY WHEEL
190     C---
191
192             INTENSITY = -1
193             CALL STANUM(INTENSITY)
194     600     CONTINUE
195
196     C---
197     C---    UPDATE DIRECTION OF WHEEL
198     C---
199
200             IDIRECT = JDIRECT
201             IF (J .GT. 0) GO TO 200
202
203             RETURN
204             END
```

```
      SUBROUTINE MODEL(TURNS,ISPOKE,Y)

      DIMENSION Y(3)

C---
C---  ROUTINE TO IMPLEMENT J.S. NORTON'S MATHEMATICAL MODEL FOR
C---  DETERMINING RIM DEFLECTIONS.
C---
      INTEGER S, SS
      DIMENSION YNINE(36)

C---
C---  F9 = FAPPLIED / 8
C---  FAPPLIED = 1.5 * TURNS OF SPOKE ADJUSTMENT
C---
      F9 = 1.5 * TURNS / 8.
      L = 72
      E = 10. ** 7
      ZI = 0.02
C---
C---  FOR S = 0 TO 9 STEP 1
C---  Y(9,S) = F9*(2*S)**3 * (3L-4(2*S)) / (48EI)
C---
      DO 100 S = 1,9
         YNINE(S) = F9 * (2*S) ** 3 * (3*L - 8*S) / E*ZI*48.

100   CONTINUE
C---
C---  FOR S = 10 TO 17 STEP 1
C---  Y(9,S) = Y(9,18-S)   BY SYMMETRY
C---
      DO 200 S = 10,17
         YNINE(S) = YNINE(18-S)

200   CONTINUE
C---
C---  FOR S = 18 TO 36 STEP 1
C---  Y(9,S) = 0
C---
      DO 300 S = 18,36
         YNINE(S) = 0.

300   CONTINUE
C---
C---  FOR S = K-8 TO K STEP 1
C---  Y(K,S) = Y(9,S-(K-9))
C---
      ITEMP = ISPOKE - 8
      DO 400 SS = ITEMP, ISPOKE
         S = SS
         IF (S .LE. 0) S = S + 36
         Y(S) = YNINE(SS - (ISPOKE - 9))
         Y(S) = Y(S) + 0.1 * Y(S) * (RANDOM(1) - 0.5)

400   CONTINUE

```

```
C---
C---    FOR S = K+1 TO K+8 STEP 1
C---    Y(K,S) = Y(9,K+9-S)
C---
        ITEMP = ISPOKE + 1
        JTEMP = ISPOKE + 8
        DO 500 SS = ITEMP, JTEMP
           S = SS
           IF (S .GT. 36) S = S - 36
           Y(S) = YNINE(ISPOKE + 9 - SS)

500     CONTINUE
C---
C---    FOR S = K+9 TO K+17 STEP 1
C---    Y(K,S) = 0
C---
        ITEMP = ISPOKE + 9
        JTEMP = ISPOKE + 17
        DO 600 SS = ITEMP,JTEMP
           S = SS
           IF (S .GT. 36) S = S - 36
           Y(S) = 0.

600     CONTINUE
C---
C---    FOR S = K-18 TO K-9 STEP 1
C---    Y(K,S) = 0
C---
        ITEMP = ISPOKE - 18
        JTEMP = ISPOKE - 9
        DO 700 SS = ITEMP,JTEMP
           S = SS
           IF (S .LE. 0) S = S + 36
           Y(S) = 0.

700     CONTINUE

        RETURN
        END
```

```
      SUBROUTINE MENU

C---
C---  ROUTINE TO OUTPUT USER OPTION MENU
C---

      TYPE "<NL><NL>   A : ADJUST SPOKES "
      TYPE "    C : ADJUST CALIPERS "
      TYPE "    D : CHANGE DIRECTION OF WHEEL MOVEMENT "
X     TYPE "    M : MEASURE WHEEL VARIANCES "
X     TYPE "    Q : QUIT "
      TYPE "    S : STOP WHEEL "
      TYPE "    T : TURN WHEEL "
      TYPE "    W : CHANGE SPEED OF WHEEL "

      RETURN
      END
```

```
      SUBROUTINE ADJUST

C---
C---  ROUTINE TO ADJUST SPOKES
C---

      COMMON /STP/ ISTOP
      COMMON /PTS/ NPOINTS(36),NDATA(180,3)
      COMMON /SPKS/ SPNUM(5),NSTR(36,2)
      COMMON /YLOC/ YCOOR(5)
      COMMON /ILIR/ IL,IR
      COMMON /LRC/ LEFT,RIGHT,CENTER
      COMMON /IPT/ IPOINT
      COMMON /CNT/ J
      COMMON /CAL/ ILEFT,IRIGHT
      COMMON /MOD/ YS(36)

      DIMENSION YY(36),INPUT(3)

      INTEGER CENTER,YCOOR,RIGHT,SPNUM

C---
C---  CHECK IF WHEEL IS STOPPED
C---

      IF (ISTOP .EQ. 0) GO TO 10
      TYPE "<BEL>"
      TYPE "<NL><NL>*****************************************************************"
      TYPE "*** YOU MUST STOP THE WHFEL BEFORE ADJUSTING SPOKES ***"
      TYPE "*****************************************************************"
      GO TO 90

C---
C---  PROMPT USER FOR SPOKE NUMBER
C---

10    CONTINUE
      TYPE "<NL><NL>WHICH SPOKE DO YOU WANT TO ADJUST  ? "
      WRITE(10,1000)
1000  FORMAT ("<NL>TYPE 1, 2, 3, ..., OR 36 ( S  TO STOP SPOKE ADJUSTMENT) :  ",Z)
      READ(11,2000) (INPUT(I), I = 1,5)
2000  FORMAT(3R1)
      IF (INPUT(1) .EQ. " S") GO TO 90
      CALL CONVERT(INPUT,ISPOKE)
      IF ((ISPOKE .LT. 0) .OR. (ISPOKE .GT. 36)) GO TO 10
      IMOD = MOD(ISPOKF,2)

C---
C---  PROMPT USER FOR DIRECTION OF SPOKF ADJUSTMENT
C---

16    CONTINUE
      TYPE "<NL><NL>   1: TURN SPOKE CLOCKWISE"
      TYPE "   2: TURN SPOKE COUNTER-CLOCKWISE"
      TYPE "   S: STOP ADJUSTMENTS ON SPOKF",ISPOKF
      WRITE(10,4000)
```

```
      4000    FORMAT("<NL>TYPE 1, 2, OR S : ",Z)
              READ(11,3000) ICHAR
      3000    FORMAT(R1)
              IF (ICHAR .EQ. " S") GO TO 10
              IF ((ICHAR .NE. " 1") .AND. (ICHAR .NE. " 2")) GO TO 16
              IDIR = 1
              IF (ICHAR .EQ. " 2") IDIR = 2
              TURNS = 1./8.
              IF ((IDIR .EQ. 2) .AND. (IMOD .EQ. 1)) TURNS = -TURNS
              IF ((IDIR .EQ. 1) .AND. (IMOD .EQ. 0)) TURNS = -TURNS

      C---
      C---    DETERMINE WHEEL VARIANCES WITH J. NORTON'S MODEL
      C---

              CALL MODEL(TURNS,ISPOKE,YY)
              DO 13 I = 1,36
                 YS(I) = YS(I) + YY(I)

      13      CONTINUE

      C---
      C---    DETERMINE NUMBER OF PIXELS WHEEL IS OFF CENTER AT EACH SPOKE
      C---

              DO 20 I = 1,36
                 NPOINTS(I) = YS(I) * 1000 + .5

      20      CONTINUE
              CALL UPDAT

      C---
      C---    ERASE OLD SPOKES
      C---

              DO 40 I = 1,5
                 CALL DSCIR(CENTER,YCOOR(I),2,0)
                 CALL DSSAO(CENTER-9,YCOOR(I)+4,0,0,1)
                 ITEMP = 6-I+J/5
                 IF (ITEMP .GT. 36) ITEMP = ITEMP-36
                 CALL DSTXT(NSTR(ITEMP,1))
                 CALL DSTXT(NSTR(ITEMP,2))
      40      CONTINUE

      C---
      C---    CHECK IF WHEEL IS HITTING THE CALIPERS
      C---

              IL = LEFT
              IR = RIGHT
              IC = CENTER
              LEFT = NDATA(IPOINT,1)
              RIGHT = NDATA(IPOINT,2)
              CENTER = NDATA(IPOINT,3)
              IF ((LEFT .GT. ILEFT) .AND. (RIGHT .LT. IRIGHT)) GO TO 60
              IF (LEFT .GT. ILEFT) GO TO 50
```

```
113             LEFT = ILEFT + 1
114             RIGHT = LEFT + 50
115             GO TO 55
116
117     50      CONTINUE
118             RIGHT = IRIGHT -1
119             LEFT = RIGHT - 50
120
121     55      CONTINUE
122             CENTER = (LEFT + RIGHT) /2
123             TYPE "<BEL>"
124             ISTOP = 0
125
126     60      CONTINUE
127
128     C---
129     C---    ERASE OLD POSITION OF THE WHEEL
130     C---
131
132             CALL DSVEC(IL,56,IL,256,0)
133             CALL DSVEC(IR,56,IR,256,0)
134
135     C---
136     C---    DISPLAY NEW POSITION OF WHEEL
137     C---
138
139             CALL DSVEC(LEFT,56,LEFT,256,-1)
140             CALL DSVEC(RIGHT,56,RIGHT,256,-1)
141
142     C---
143     C---    DISPLAY SPOKE NUMBERS
144     C---
145
146             DO 80 I = 1,5
147                IF ((YCOOR(I) .LT. 56) .OR. (YCOOR (I) .GT. 250)) GO TO 80
148                CALL DSCIR(CENTER,YCOOR(I),2,-1)
149                CALL DSSAO(CENTER-9,YCOOR(I)+4,-1,0,1)
150                ITEMP = 6 - I + J/5
151                IF (ITEMP .GT. 36) ITEMP = ITEMP - 36
152                CALL DSTXT(NSTR(ITEMP,1))
153                CALL DSTXT(NSTR(ITEMP,2))
154
155     80      CONTINUE
156             GO TO 16
157     90      CONTINUE
158
159             RETURN
160             END
```

```
      SUBROUTINE ADJCAL

C---
C---  ROUTINE TO ADJUST CALIPERS
C---

      INTEGER CALEFT,CALRGT

      COMMON /CAL/ ILEFT,IRIGHT
      COMMON /ILIR/ IL,IR

C---
C---  PROMPT USER FOR ADJUSTMENT
C---

4     CONTINUE
      CALEFT = 1
      CALRGT = -1
      TYPE "<NL><NL>   I : MOVE CALIPERS IN "
      TYPE "   O : MOVE CALIPERS OUT "
      TYPE "   S : STOP CALIPER ADJUSTMENT "
      WRITE(10,5000)
5000  FORMAT("<NL>TYPE I, O, OR S : ",Z)
      READ(11,6000) ICHAR
6000  FORMAT(R1)
      IF (ICHAR .EQ. " 0") GO TO 10
      ICHAR = ICHAR - 20100K
      IF ((ICHAR .LT. 1) .OR. (ICHAR .GT. 26)) GO TO 4
      GO TO (4,4,4,4,4,4,4,4,20,4,4,4,4,4,10,4,4,4,30,4,4,4,4,4,4,4),ICHAR

16    CONTINUE
      CALEFT = -1
      CALRGT = 1

C---
C---  CHECK FOR ILLEGAL ADJUSTMENT OF CALIPERS
C---

20    CONTINUE
      ITEMP = ILEFT
      JTEMP = IRIGHT
      DO 22 I = 1,4
         ITEMP = ITEMP + CALEFT
         JTEMP = JTEMP + CALRGT
         IF ((IL .GE. ITEMP) .AND. (IR .LE. JTEMP) .AND.
     +      (ITEMP .GE. 25) .AND. (ITEMP .LE. 231)) GO TO 22

23       CONTINUE
         ITEMP = ITEMP - CALEFT
         JTEMP = JTEMP - CALRGT
         ISTOP = 0
         TYPE "<BEL>"
         GO TO 25

22    CONTINUE
```

```
   57      25      CONTINUE
   58
   59      C---
   60      C---    ERASE OLD POSITION OF CALIPERS
   61      C---
   62
   63              INTEN = 0
   64              CALL CALIP(ILEFT,IRIGHT,INTEN)
   65
   66      C---
   67      C---    DISPLAY NEW POSITION OF CALIPERS
   68      C---
   69
   70              ILEFT = ITEMP
   71              IRIGHT = JTEMP
   72              INTEN = -1
   73              CALL CALIP(ILEFT,IRIGHT,INTEN)
   74              GO TO 4
   75
   76      50      CONTINUE
   77              RETURN
   78              END
```

```
      SUBROUTINE CALIP(ILEFT,IRIGHT,INTEN)

C---
C---  ROUTINE TO DRAW OR ERASE CALIPERS
C---

      ITEMP = ILEFT - 22
      JTEMP = IRIGHT + 22

      CALL DSVEC(ILEFT,135,ITEMP,123,INTEN)
      CALL DSVEC(ILEFT,135,ITEMP,147,INTEN)
      CALL DSVEC(ITEMP,123,ITEMP,147,INTEN)

      CALL DSVEC(IRIGHT,135,JTEMP,123,INTEN)
      CALL DSVEC(IRIGHT,135,JTEMP,147,INTEN)
      CALL DSVEC(JTEMP,123,JTEMP,147,INTEN)

      RETURN
      END
```

CC

```
      1            SUBROUTINE UPDAT
      2
      3      C
      4      C     ROUTINE TO UPDATE THE POSITION OF WHEEL AT EACH POINT ALONG A REVOLUTION
      5      C
      6
      7            COMMON /PTS/ NPOINTS(36),NDATA(180,3)
      8            DIMENSION IPIX(5)
      9
     10               DO 17 I = 1,36
     11                  LAST = I - 1
     12                  IF (LAST .EQ. 0) LAST = 36
     13                  DO 14 K = 1,5
     14                     IPIX(K) = (NPOINTS(I) - NPOINTS(LAST)) / 5
     15
     16     14           CONTINUE
     17                  KTEMP = MOD((NPOINTS(I) - NPOINTS(LAST)),5)
     18                  IF (KTEMP .NE. 0) CALL REFINE(KTEMP,IPIX)
     19                  DO 15 K = 1,5
     20                     NPTR = (I-1) * 5 + K
     21                     IOVER = NPOINTS(LAST)
     22                     DO 18 L = 1,K
     23                        IOVER = IOVER + IPIX(L)
     24
     25     18              CONTINUE
     26                     LEFT = NDATA(NPTR,1) = IOVER + 231
     27                     RIGHT = NDATA(NPTR,2) = LEFT + 50
     28                     CENTER = NDATA(NPTR,3) = (LEFT + RIGHT) / 2
     29
     30     15           CONTINUE
     31
     32     17     CONTINUE
     33            RETURN
     34            END
```

```
      SUBROUTINE STANUM(INTEN)

C---
C---  ROUTINE TO DIPLAY SPOKE NUMBERS ON STATIONARY WHEEL
C---

      INTEGER XCOOR(5),YCOOR(5),SPNUM

      COMMON /SPKS/ SPNUM(5),NSTR(36,2)

      DATA XCOOR/446,415,381,346,314/
      DATA YCOOR/428,448,458,456,444/
X     DATA XCOOR/314,346,381,415,446/
X     DATA YCOOR/444,456,458,448,428/

      DO 200 I = 1,5
         CALL DSSAO(XCOOR(I),YCOOR(I),INTEN,0,1)
         CALL DSTXT(NSTR(SPNUM(I),1))
         CALL DSTXT(NSTR(SPNUM(I),2))
200   CONTINUE

      RETURN
      END
```

```
      SUBROUTINE REFINE(KTEMP,IPIX)

C---
C---  ROUTINE TO REFINE WHEEL MOVEMENT
C---

      DIMENSION IPIX(5)

      IONE = 1
      IF (KTEMP .LT. 0) IONE = -1
      GO TO (1,2,3,4), ABS(KTEMP)

1     CONTINUE
      IPIX(3) = IPIX(3) + IONE
      GO TO 10

2     CONTINUE
      IPIX(2) = IPIX(2) + IONE
      IPIX(4) = IPIX(4) + IONE
      GO TO 10

3     CONTINUE
      IPIX(2) = IPIX(2) + IONE
      IPIX(4) = IPIX(4) + IONE
      IPIX(5) = IPIX(5) + IONE
      GO TO 10

4     CONTINUE
      DO 5 I = 1,4
         IPIX(I) = IPIX(I) + IONE

5     CONTINUE

10    RETURN
      END
```

```
      SUBROUTINE CONVERT(INPUT,ISPOKE)

C---
C---  ROUTINE TO CONVERT CHARACTER INPUT TO INTEGER
C---

      DIMENSION INPUT(3)

      ISPOKE = -1
      IF ((INPUT(1) .LT. 020060K) .OR. (INPUT(1) .GT. 020071K)
     + .OR. (INPUT(2) .LT. 020040K) .OR. (INPUT(2) .GT. 020071K)
     + .OR. (INPUT(3) .NE. 020040K)) GO TO 100

      ISPOKE = INPUT(1) - 020060K
      IF (INPUT(2) .LT. 020060K) GO TO 100
      ISPOKE = ISPOKE * 10 + INPUT(2) - 020060K

100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PWHEEL

C---
C---  THIS ROUTINE DRAWS A PORTION OF A BICYCLE WHEEL ON
C---  THE LOWER RIGHT CORNER OF THE MONITOR
C---

      DIMENSION IX(8),IY(8),JX(8),JY(8),KX(20),KY(20),IRAD(3),OUT(1000)

      INTEGER OUT

      DATA IRAD/105,121,136/

      NPTS = 1000
      DEL = 3.14159 * 2. / NPTS
      IXCENT = 375
      IYCENT = 350
      DO 400 I = 1,3
         IRADIUS = IRAD(I)
         DO 300 J = 115,365
            OUT((J-114) * 2 - 1) = IXCENT + IRADIUS * COS(J * DEL)
            OUT((J-114) * 2) = IYCENT + IRADIUS * SIN(J * DEL)

300      CONTINUE
         CALL DSPNT(250,-1,OUT)

400   CONTINUE

C---
C---  DISPLAY SPOKE CIRCLES ON WHEEL
C---

      KRADIUS = 112
      NPTS = 20
      DEL = 3.14159 * 2. / NPTS
      KXCENT = 375
      KYCENT = 350
      DO 500 I = 1,NPTS
         KX(I) = KXCENT + KRADIUS * COS(I * DEL) + 2
         KY(I) = KYCENT + KRADIUS * SIN(I * DEL)

500   CONTINUE

      DO 600 I = 3,7
         CALL DSCIR(KX(I),KY(I),2,-1)
         CALL DSCIR(KX(I),KY(I),1,-1)

600   CONTINUE

C---
C---  DISPLAY SHADING ON WHEEL
C---

      CALL DSVEC(373,478,400,478,-1)
      CALL DSVEC(374,480,380,480,-1)
      CALL DSVEC(390,482,395,482,-1)
```

```
      CALL DSVEC(335,473,350,473,-1)
      CALL DSVEC(340,475,345,475,-1)

      CALL DSVEC(300,460,310,460,-1)

      CALL DSVEC(425,467,435,467,-1)

C---
C---  DISPLAY HUB
C---

      CALL DSCIR(375,320,15,-1)
      CALL DSCIR(375,320,3,-1)
      CALL DSCIR(375,290,15,-1)

C---
C---  DISPLAY SPOKE CIRCLES ON HUB
C---

      NPTS = 8
      DEL = 3.14159 * 2. / NPTS
      IRADIUS = 12
      JRADIUS = 12
      IXCENT = 375
      IYCENT = 320
      JXCENT = 375
      JYCENT = 290
      IOFF = 0
      DO 100 I = 1,NPTS
         IX(I) = IXCENT + IRADIUS * COS(I * DEL + IOFF)
         IY(I) = IYCENT + IRADIUS * SIN(I * DEL + IOFF)
         JX(I) = JXCENT + JRADIUS * COS(I * DEL + IOFF)
         JY(I) = JYCENT + JRADIUS * SIN(I * DEL + IOFF)
         CALL DSCIR(IX(I),IY(I),1,-1)
         IF (I .NE. 2) CALL DSCIR(JX(I),JY(I),1,-1)

100   CONTINUE
      CALL DSCIR(375,290,3,-1)
      DO 200 I = 372,378
         CALL DSVEC(I,290,I,304,0)

200   CONTINUE
      CALL DSVEC(372,290,372,305,-1)
      CALL DSVEC(378,290,378,305,-1)

C---
C---  DISPLAY SPOKES
C---

      CALL DSVEC(JX(8),JY(8),JX(8)+40,JY(8)+50,-1)
      CALL DSVEC(JX(1),JY(1),KX(3),KY(3),-1)
      CALL DSVEC(IX(8),IY(8),IX(8)+50,IY(8)+75,-1)
      CALL DSVEC(IX(1),IY(1),KX(4),KY(4),-1)
      CALL DSVEC(IX(2)+2,IY(2)+3,KX(5),KY(5),-1)
      CALL DSVEC(IX(2),IY(2),KX(6),KY(6),-1)
      CALL DSVEC(IX(3),IY(3),IX(3)-50,IY(3)+60,-1)
      CALL DSVEC(JX(3),JY(3),KX(7),KY(7),-1)
```

```
113             CALL DSVEC(JX(4),JY(4),JX(4)-40,JY(4)+50,-1)
114             RETURN
115             END
```

```
      1              SUBROUTINE DEMOS
      2
      3      C---
      4      C---    ROUTINE TO DISPLAY INSTRUCTOR DEMOS
      5      C---
      6
      7              INTEGER DEMO
      8
      9              TYPE "<NL>   1: EQUIPMENT COMPONENTS "
     10              TYPE "   2: FINDING WOBBLE"
     11              TYPE "   3: SPOKE ADJUSTMENT"
     12              TYPE "   4: FINE TUNING"
     13
     14      10      CONTINUE
     15              WRITE (10,1000)
     16      1000    FORMAT("<NL>TYPE 1, 2, 3, 4  (0 TO STOP DEMOS) : ",Z)
     17              READ (11,2000) DEMO
     18      2000    FORMAT(R1)
     19
     20      C---
     21      C---    CLEAR SCREEN
     22      C---
     23              CALL DSCLR(-1)
     24
     25              IF (DEMO .EQ. " 0") GO TO 60
     26              IF (DEMO .EQ. " 1") GO TO 20
     27              IF (DEMO .EQ. " 2") GO TO 30
     28              IF (DEMO .EQ. " 3") GO TO 40
     29              IF (DEMO .EQ. " 4") GO TO 50
     30              GO TO 10
     31
     32      20      CALL DEMO1
     33              GO TO 10
     34
     35      30      CALL DEMO2
     36              GO TO 10
     37
     38      40      CALL DEMO3
     39              GO TO 10
     40
     41      50      CALL DEMO4
     42              GO TO 10
     43
     44      60      CONTINUE
     45              RETURN
     46              END
```

```
      SUBROUTINE DEMO1

C---
C---  ROUTINE TO DISPLAY INSTRUCTOR DEMO 1 (EQUIPMENT COMPONENTS)
C---

      DIMENSION IX(8),IY(8),KX(8),KY(8),LX(16),LY(16),MX(8),MY(8)
      DIMENSION NX(8),NY(8)

      INTEGER WDSIZ,X(8),Y(8)

C---
C---  DISPLAY WHEEL AND HUB
C---

      CALL DSCIR(256,256,5,-1)
      CALL DSCIR(256,256,10,-1)
      CALL DSCIR(256,256,35,-1)
      CALL DSCIR(256,256,115,-1)
      CALL DSCIR(256,256,130,-1)

C---
C---  DISPLAY TRUEING STAND AND CALIPERS
C---

      CALL DSVEC(253,291,250,371,-1)
      CALL DSVEC(259,291,262,371,-1)
      CALL DSVEC(249,386,249,400,-1)
      CALL DSVEC(263,386,263,400,-1)
      CALL DSVEC(100,400,412,400,-1)
      CALL DSVEC(100,403,412,403,-1)
      CALL DSVEC(100,400,100,403,-1)
      CALL DSVEC(412,400,412,403,-1)
      CALL DSVEC(179,339,164,354,-1)
      CALL DSVEC(177,337,162,352,-1)
      CALL DSVEC(179,339,177,337,-1)
      CALL DSVEC(159,349,167,357,-1)
      CALL DSVEC(159,349,151,357,-1)
      CALL DSVEC(167,357,159,365,-1)
      CALL DSVEC(151,357,159,365,-1)
      CALL DSVEC(153,359,151,361,-1)
      CALL DSVEC(157,363,155,365,-1)
      CALL DSVEC(147,357,159,369,-1)
      CALL DSVEC(144,360,156,372,-1)
      CALL DSVEC(147,357,144,360,-1)
      CALL DSVEC(159,369,156,372,-1)

      IX1 = 147
      IY1 = 357
      IX2 = 144
      IY2 = 360
      DO 50 I = 1,5
         IX1 = IX1 + 2
         IY1 = IY1 + 2
         IX2 = IX2 + 2
         IY2 = IY2 + 2
```

```
      CALL DSVEC(IX1,IY1,IX2,IY2,-1)

50    CONTINUE

C---
C---  DISPLAY SPOKES AND SPOKE POINTS ON HUB
C---

      NPTS = 8
      DEL = 3.14159 * 2. / NPTS
      IRADIUS = 28
      JRADIUS = 115
      KRADIUS = 35
      LRADIUS = 113
      OFF = 1
      OFF1 = -1
      DO 100 I = 1,NPTS
         X(I) = 256 + IRADIUS * COS(I * DEL)
         Y(I) = 256 + IRADIUS * SIN(I * DEL)
         CALL DSCIR(X(I),Y(I),2,-1)
100   CONTINUE

      DO 200 I = 1,NPTS
         IX(I) = 256 + JRADIUS * COS(I * DEL + OFF)
         IY(I) = 256 + JRADIUS * SIN(I * DEL + OFF)
         MX(I) = 256 + LRADIUS * COS(I * DEL + OFF)
         MY(I) = 256 + LRADIUS * SIN(I * DEL + OFF)
         CALL DSVEC(X(I),Y(I),IX(I),IY(I),-1)

200   CONTINUE

      NPTS2 = 16
      DEL2 = 3.14159 * 2. / NPTS2
      DO 250 I = 1,NPTS2
         LX(I) = 256 + KRADIUS * COS(I * DEL2)
         LY(I) = 256 + KRADIUS * SIN(I * DEL2)

250   CONTINUE
      DO 300 I = 1,NPTS
         KX(I) = 256 + JRADIUS * COS(I * DEL + OFF1)
         KY(I) = 256 + JRADIUS * SIN(I * DEL + OFF1)
         NX(I) = 256 + LRADIUS * COS(I * DEL + OFF1)
         NY(I) = 256 + LRADIUS * SIN(I * DEL + OFF1)
         CALL DSVEC(LX(I*2-1),LY(I*2-1),KX(I),KY(I),-1)

300   CONTINUE

C---
C---  DISPLAY SPOKE CIRCLES ON WHEEL
C---

      DO 400 I = 1,NPTS
         CALL DSCIR(MX(I),MY(I),2,-1)
         CALL DSCIR(NX(I),NY(I),2,-1)
         CALL DSCIR(MX(I),MY(I),1,-1)
         CALL DSCIR(NX(I),NY(I),1,-1)
```

```
113
114     400     CONTINUE
115
116
117     C---
118     C---    DISPLAY WRENCH
119     C---
120
121             CALL #RENCH
122
123     C---
124     C---    DISPLAY TITLE
125     C---
126
127             CALL DSSAO(175,450,-1,0,1)
128             CALL DSTXT("1. EQUIPMENT COMPONENTS")
129
130             RETURN
131             END
```

```
      SUBROUTINE DEMO2

C---
C---  ROUTINE TO DISPLAY INSTRUCTOR DEMO 2 (FINDING WOBBLE)
C---

      DIMENSION IX(8),IY(8),JX(8),JY(8),KX(16),KY(16),IRAD(3),OUT(1000)

      INTEGER OUT

      DATA IRAD/110,126,141/


C---
C---  DISPLAY RIM
C---

      NPTS = 1500
      DEL = 3.14159 * 2. / NPTS
      IXCENT = 256
      IYCENT = 256
      DO 200 I = 1,3
         IRADIUS = IRAD(I)
         DO 100 J = 125,625
            OUT((J-124) * 2 - 1) = IXCENT + IRADIUS * COS(J * DEL)
            OUT((J-124) * 2) = IYCENT + IRADIUS * SIN(J * DEL)

  100    CONTINUE
         CALL DSPNT(500,-1,OUT)

  200 CONTINUE

C---
C---  DISPLAY CIRCLES ON RIM
C---

      KRADIUS = 117
      NPTS = 16
      DEL = 3.14159 * 2. / NPTS
      KXCENT = 256
      KYCENT = 256
      DO 300 I = 1,NPTS
         KX(I) = KXCENT + KRADIUS * COS(I * DEL) + 2
         KY(I) = KYCENT + KRADIUS * SIN(I * DEL)

  300 CONTINUE

      DO 400 I = 2,6
         CALL DSCIR(KX(I),KY(I),2,-1)
         CALL DSCIR(KX(I),KY(I),1,-1)

  400 CONTINUE
C---
C---  DISPLAY SPOKE NUMBERS
C---

```

```
      CALL DSSAO(KX(6)+4,KY(6)+4,-1,0,1)
      CALL DSTXT("1")
      CALL DSSAO(KX(5)+5,KY(5),-1,0,1)
      CALL DSTXT("2")
      CALL DSSAO(KX(4)+5,KY(4)-3,-1,0,1)
      CALL DSTXT("3")
      CALL DSSAO(KX(3)+5,KY(3)-7,-1,0,1)
      CALL DSTXT("4")
      CALL DSSAO(KX(2)+5,KY(2)-10,-1,0,1)
      CALL DSTXT("5")

C---
C---  ADD SHADING TO RIM
C---

      CALL DSVEC(254,387,281,387,-1)
      CALL DSVEC(255,388,261,388,-1)
      CALL DSVEC(271,392,276,392,-1)
      CALL DSVEC(216,382,231,382,-1)
      CALL DSVEC(221,384,226,384,-1)
      CALL DSVEC(187,373,197,373,-1)
      CALL DSVEC(306,376,316,376,-1)
C---
C---  DISPLAY HUB
C---

      CALL DSCIR(256,150,20,-1)
      CALL DSCIR(256,110,20,-1)
      CALL DSCIR(256,150,4,-1)
      CALL DSCIR(256,110,4,-1)
      DO 350 I = 252,260
         CALL DSVEC(I,110,I,116,0)

350   CONTINUE
      CALL DSVEC(252,110,252,130,-1)
      CALL DSVEC(260,110,260,130,-1)

C---
C---  DISPLAY SPOKE CIRCLES ON HUB
C---

      NPTS = 8
      DEL = 3.14159 * 2. / NPTS
      IRADIUS = 16
      JRADIUS = 16
      IXCENT = 256
      IYCENT = 150
      JXCENT = 256
      JYCENT = 110
      DO 500 I = 1,NPTS
         IX(I) = IXCENT + IRADIUS * COS(I * DEL)
         IY(I) = IYCENT + JRADIUS * SIN(I * DEL)
         JX(I) = JXCENT + IRADIUS * COS(I * DEL)
         JY(I) = JYCENT + JRADIUS * SIN(I * DEL)
         CALL DSCIR(IX(I),IY(I),1,-1)
         IF (I .NE. 2) CALL DSCIR(JX(I),JY(I),1,-1)
```

```
113
114      500     CONTINUE
115      C---
116      C---     DISPLAY SPOKES
117      C---
118              CALL DSVEC(JX(8),JY(8),JX(8)+40,JY(8)+60,-1)
119              CALL DSVEC(JX(1),JY(1),KX(2),KY(2),-1)
120              CALL DSVEC(IX(8),IY(8),IX(8)+50,IY(8)+100,-1)
121              CALL DSVEC(IX(1),IY(1),KX(3),KY(3),-1)
122              CALL DSVEC(IX(2)+2,IY(2)+6,KX(4),KY(4),-1)
123              CALL DSVEC(IX(2),IY(2),KX(5),KY(5),-1)
124              CALL DSVEC(IX(3),IY(3),IX(3)-45,IY(3)+75,-1)
125              CALL DSVEC(JX(3),JY(3),KX(6),KY(6),-1)
126              CALL DSVEC(JX(4),JY(4),JX(4)-40,JY(4)+65,-1)
127
128      C---
129      C---     DISPLAY CALIPERS
130      C---
131
132              CALL DSVEC(168,377,183,388,-1)
133              CALL DSVEC(168,377,183,362,-1)
134              CALL DSVEC(183,388,183,362,-1)
135              DO 700 I = 168,183
136                 CALL DSVEC(I,545-I,183,388,-1)
137
138      700     CONTINUE
139
140              CALL DSVEC(193,330,193,304,-1)
141              CALL DSVEC(193,330,208,315,-1)
142              CALL DSVEC(193,304,208,315,-1)
143              DO 800 I = 193,208
144                 CALL DSVEC(I,523-I,193,304,-1)
145
146      800     CONTINUE
147
148      C---
149      C---     DISPLAY TITLE
150      C---
151
152              CALL DSSAD(200,450,-1,0,1)
153              CALL DSTXT("2. FINDING WOBBLE")
154
155              RETURN
156              END
```

```
      SUBROUTINE DEMO3

C---
C---  ROUTINE TO DISPLAY INSTRUCTOR DEMO 3 (SPOKE ADJUSTMENT)
C---

      DIMENSION OUT(804),IRAD(6)
      INTEGER OUT

      DATA IRAD/140,155,157,183,193,195/

C---
C---  DISPLAY DIRECTIONAL ARROWS
C---

      DO 5 I = 7,9
         CALL DSCIR(210,203,I,-1)
         CALL DSCIR(256,159,I,-1)

5     CONTINUE

      DO 6 J = 1,10
         CALL DSVEC(150+J,500,250+J,0,0)
         CALL DSVEC(330+J,500,205+J,0,0)

6     CONTINUE
      DO 8 I = 192,198
         CALL DSVEC(212,I,218,195,-1)

8     CONTINUE
      DO 9 I = 148,154
         CALL DSVEC(253,I,248,151,-1)

9     CONTINUE

C---
C---  DISPLAY RIM
C---

      NPTS = 2000
      DEL = 3.14159 * 2. /NPTS
      IXCENT = 350
      IYCENT = 300
      DO 30 I = 1,6
         IRADIUS = IRAD(I)
         DO 10 J = 1000,1400
            OUT((J-999) * 2 - 1) = IXCENT + IRADIUS * COS(J * DEL)
            OUT((J-999) * 2) = IYCENT + IRADIUS * SIN(J * DEL)

10       CONTINUE

         CALL DSPNT(400,-1,OUT)

30    CONTINUE

C---
```

```
C---    DISPLAY SPOKE CIRCLES ON RIM
C---
        CALL DSCIR(185,275,1,-1)
        CALL DSCIR(185,275,3,-1)
        CALL DSCIR(210,203,1,-1)
        CALL DSCIR(210,203,3,-1)
        CALL DSCIR(256,159,1,-1)
        CALL DSCIR(256,159,3,-1)

C---
C---    DISPLAY SPOKES
C---

        CALL DSVEC(211,278,300,310,-1)
        CALL DSVEC(235,218,325,300,-1)
        CALL DSVEC(275,180,345,285,-1)

C---
C---    DISPLAY TEXT
C---

        CALL DSSAD(145,195,-1,0,1)
        CALL DSTXT("TIGHTEN")
        CALL DSSAD(256,135,-1,0,1)
        CALL DSTXT("LOOSEN")

C---
C---    DISPLAY TITLE
C---

        CALL DSSAD(185,450,-1,0,1)
        CALL DSTXT("3. SPOKE ADJUSTMENT")
        RETURN
        END
```

```
      SUBROUTINE DEMO4

C---
C---  ROUTINE TO DISPLAY INSTRUCTOR DEMO 4 (FINE TUNING)
C---

      DIMENSION YCOOR(7),XCOOR(7),JSTRING(7),IX(7,8),IY(7,8)
      DIMENSION IOUT(1500),IXCENT(2)

      INTEGER YCOOR,XCOOR

      DATA XCOOR/258,252,248,245,248,252,258/
      DATA YCOOR/121,164,207,252,293,336,379/
      DATA IXCENT/451,415/

      JSTRING(7) = "1"
      JSTRING(6) = "2"
      JSTRING(5) = "3"
      JSTRING(4) = "4"
      JSTRING(3) = "5"
      JSTRING(2) = "6"
      JSTRING(1) = "7"

C---
C---  DISPLAY DOTTED LINE
C---

      DO 100 I = 100,385,15
         CALL DSVEC(280,I,280,I+10,-1)

100   CONTINUE

C---
C---  DISPLAY RIM
C---

      NPTS = 10000
      DEL = 3.14159 * 2. / NPTS
      IYCENT = 256
      IRADIUS = 700
      DO 165 J = 1,2
      IXC = IXCENT(J)
      DO 155 I = 4675,5350
         IOUT((I-4674) * 2 - 1) = IXC + IRADIUS * COS(I * DEL)
         IOUT((I-4674) * 2) = IYCENT + IRADIUS * SIN(I * DEL)

155   CONTINUE
      CALL DSPNT(675,-1,IOUT)
165   CONTINUE

C---
C---  DISPLAY SPOKE CIRCLES
C---

      NPTS = 8
      DEL = 3.14159 * 2. / NPTS
```

```
      IRADIUS = 4
      DO 200 I = 1,7
         CALL DSCIR(XCOOR(I),YCOOR(I),4,-1)
         CALL DSCIR(XCOOR(I),YCOOR(I),1,-1)
         CALL DSSAO(XCOOR(I)-3,YCOOR(I)+6,-1,0,1)
         CALL DSTXT(JSTRING(I))
         DO 150 J = 1,NPTS
            IX(I,J) = XCOOR(I) + IRADIUS * COS(J * DEL)
            IY(I,J) = YCOOR(I) + IRADIUS * SIN(J * DEL)

  150    CONTINUE

  200 CONTINUE
      CALL DSVEC(IX(1,1),IY(1,1),IX(1,5),IY(1,5),-1)
      CALL DSVEC(IX(2,3),IY(2,3),IX(2,7),IY(2,7),-1)
      CALL DSVEC(IX(3,4),IY(3,4),IX(3,8),IY(3,8),-1)
      CALL DSVEC(IX(4,2),IY(4,2),IX(4,6),IY(4,6),-1)
      CALL DSVEC(IX(5,1),IY(5,1),IX(5,5),IY(5,5),-1)
      CALL DSVEC(IX(6,4),IY(6,4),IX(6,8),IY(6,8),-1)
      CALL DSVEC(IX(7,3),IY(7,3),IX(7,7),IY(7,7),-1)

C---
C---  DISPLAY TITLE
C---

      CALL DSSAO(210,450,-1,0,1)
      CALL DSTXT("4. FINE TUNING")

      RETURN
      END
```

```
      SUBROUTINE WRENCH

C---
C---  ROUTINE TO DISPLAY SPOKE WRENCH FOR INSTRUCTOR DEMO 1
C---

      CALL DSCIN(75,275,18,-1)
      CALL DSCIR(75,275,20,-1)
      DO 100 I = 55,70
         CALL DSVEC(I,0,I,511,0)

100   CONTINUE

      CALL DSVEC(70,255,54,267,-1)
      CALL DSVEC(70,257,54,269,-1)
      CALL DSVEC(70,295,54,283,-1)
      CALL DSVEC(70,293,54,281,-1)
      CALL DSVEC(54,267,49,267,-1)
      CALL DSVEC(49,267,49,269,-1)
      CALL DSVEC(49,281,49,283,-1)
      CALL DSVEC(48,270,50,270,-1)
      CALL DSVEC(48,280,50,280,-1)
      CALL DSVEC(48,270,48,273,-1)
      CALL DSVEC(50,270,50,273,-1)
      CALL DSVEC(48,277,48,280,-1)
      CALL DSVEC(50,277,50,280,-1)
      CALL DSVEC(47,273,47,277,-1)
      CALL DSVEC(49,273,49,277,-1)
      CALL DSVEC(47,273,49,273,-1)
      CALL DSVEC(47,277,49,277,-1)
      CALL DSVEC(54,269,49,269,-1)
      CALL DSVEC(54,281,49,281,-1)
      CALL DSVEC(54,283,49,283,-1)

      RETURN
      END
```

APPENDIX E

TRAINING DEVICE FIDELITY RATING FORMS

Rater: Brock

Date: 6/15/82

Physical Similarity

1	2	3	4	5	6	7
Looks and feels nothing like the real equipment			Is moderately like the actual equipment in appearance and feel			Looks and feels like the actual equipment

Functional Similarity

1	2	3	4	5	6	7
Does not work like actual equipment (controls, displays do not work)			Works like actual equipment in some respects (controls and displays work but without effect)			Works exactly like the actual equipment (controls, displays work with effect)

Please rate each device on the above scales.

Device			
HH	1.	Physical similarity	7
		Functional similarity	7
MM	2.	Physical similarity	4
		Functional similarity	5
LL	3.	Physical similarity	2
		Functional similarity	1
HL	4.	Physical similarity	2
		Functional similarity	5
LH	5.	Physical similarity	6
		Functional similarity	3

Rater: L. Miller

Date: 6/15/82

Physical Similarity

1	2	3	4	5	6	7
Looks and feels nothing like the real equipment			Is moderately like the actual equipment in appearance and feel			Looks and feels like the actual equipment

Functional Similarity

1	2	3	4	5	6	7
Does not work like actual equipment (controls, displays do not work)			Works like actual equipment in some respects (controls and displays work but without effect)			Works exactly like the actual equipment (controls, displays work with effect)

Please rate each device on the above scales.

Device

HH	1.	Physical similarity	7
		Functional similarity	7
MM	2.	Physical similarity	3
		Functional similarity	3
LL	3.	Physical similarity	4
		Functional similarity	4
HL	4.	Physical similarity	3
		Functional similarity	4
LH	5.	Physical similarity	6
		Functional similarity	1

Rater: Modrick

Date: 6/15/82

Physical Similarity

1	2	3	4	5	6	7
Looks and feels nothing like the real equipment			Is moderately like the actual equipment in appearance and feel			Looks and feels like the actual equipment

Functional Similarity

1	2	3	4	5	6	7
Does not work like actual equipment (controls, displays do not work)			Works like actual equipment in some respects (controls and displays work but without effect)			Works exactly like the actual equipment (controls, displays work with effect)

Please rate each device on the above scales.

Device

HH	1.	Physical similarity	7
		Functional similarity	7
MM	2.	Physical similarity	3
		Functional similarity	4
LL	3.	Physical similarity	1
		Functional similarity	1
HL	4.	Physical similarity	2
		Functional similarity	5
LH	5.	Physical similarity	4
		Functional similarity	2

Rater: Daniels

Date: 6/15/82

Physical Similarity

1	2	3	4	5	6	7
Looks and feels nothing like the real equipment			Is moderately like the actual equipment in appearance and feel			Looks and feels like the actual equipment

Functional Similarity

1	2	3	4	5	6	7
Does not work like actual equipment (controls, displays do not work)			Works like actual equipment in some respects (controls and displays work but without effect)			Works exactly like the actual equipment (controls, displays work with effect)

Please rate each device on the above scales.

Device

HH	1.	Physical similarity	7
		Functional similarity	7
MM	2.	Physical similarity	4
		Functional similarity	2
LL	3.	Physical similarity	1
		Functional similarity	1
HL	4.	Physical similarity	2
		Functional similarity	6
LH	5.	Physical similarity	6
		Functional similarity	1

APPENDIX F

ILLUSTRATIONS OF DEVICE HL
(COMPUTER GRAPHICS)

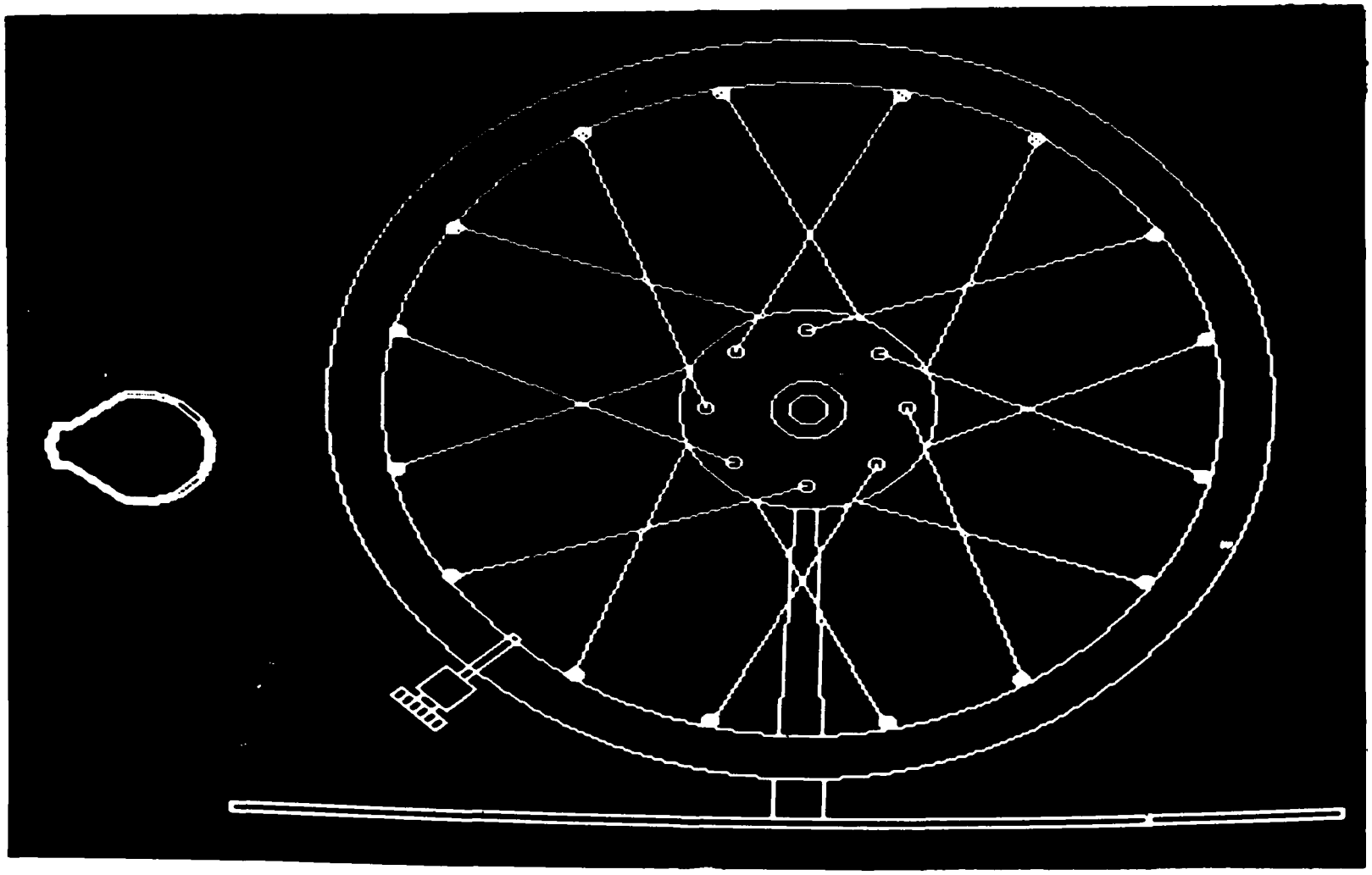


Figure F-1 Equipment components--graphics display device.

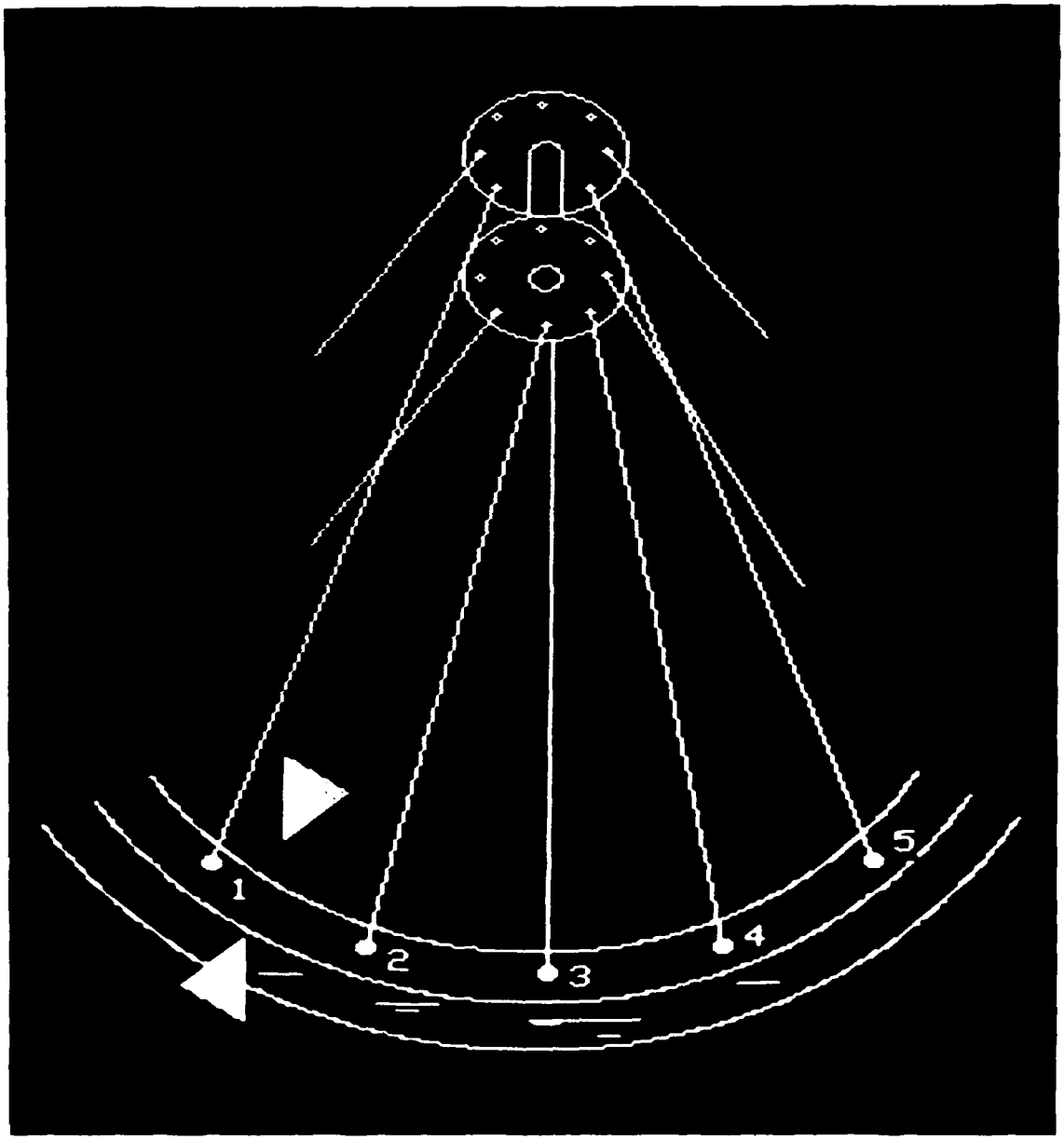


Figure F-2. Finding wobble--graphics display device.

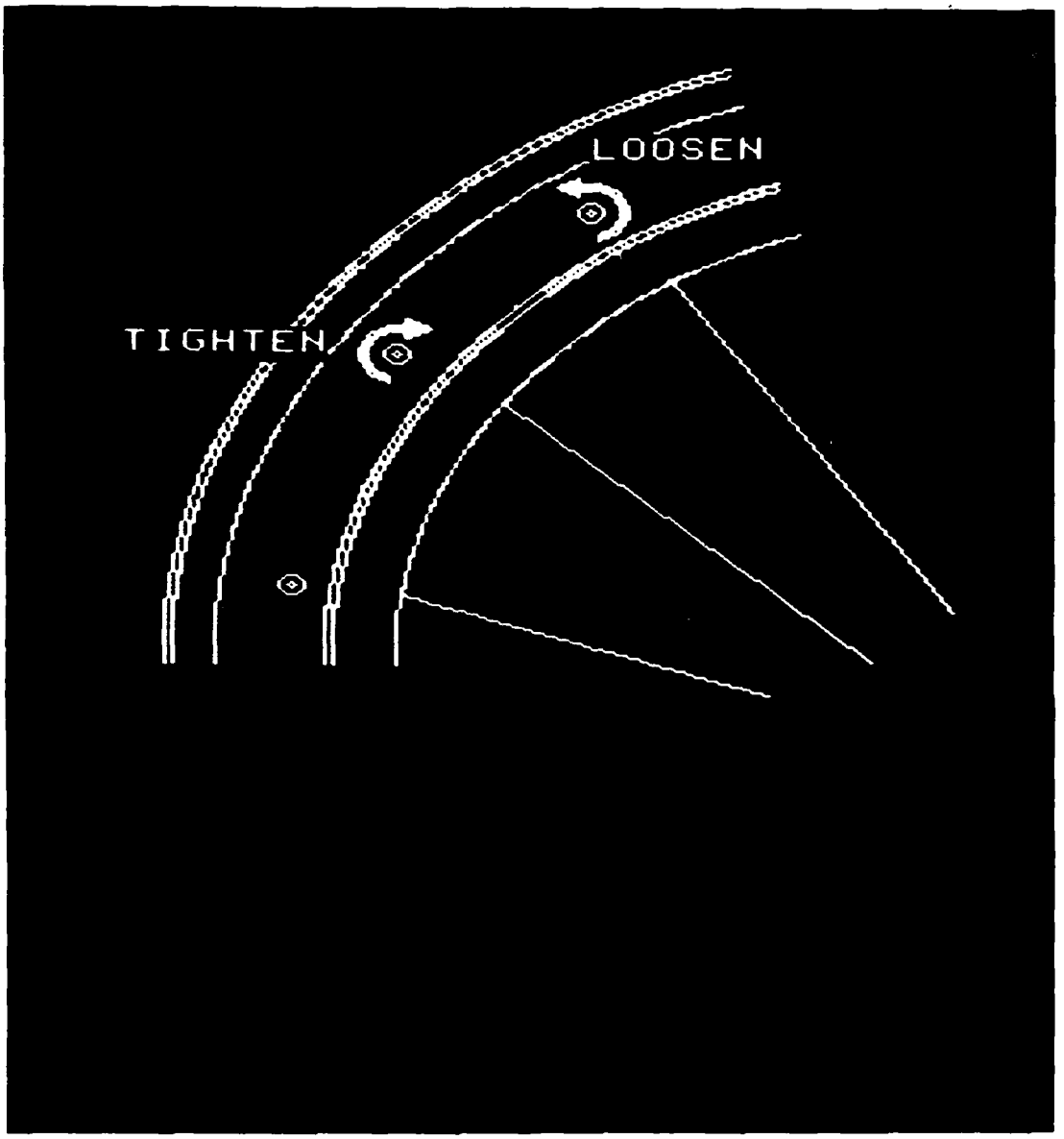


Figure F-3. Spoke adjustment--graphics display device.

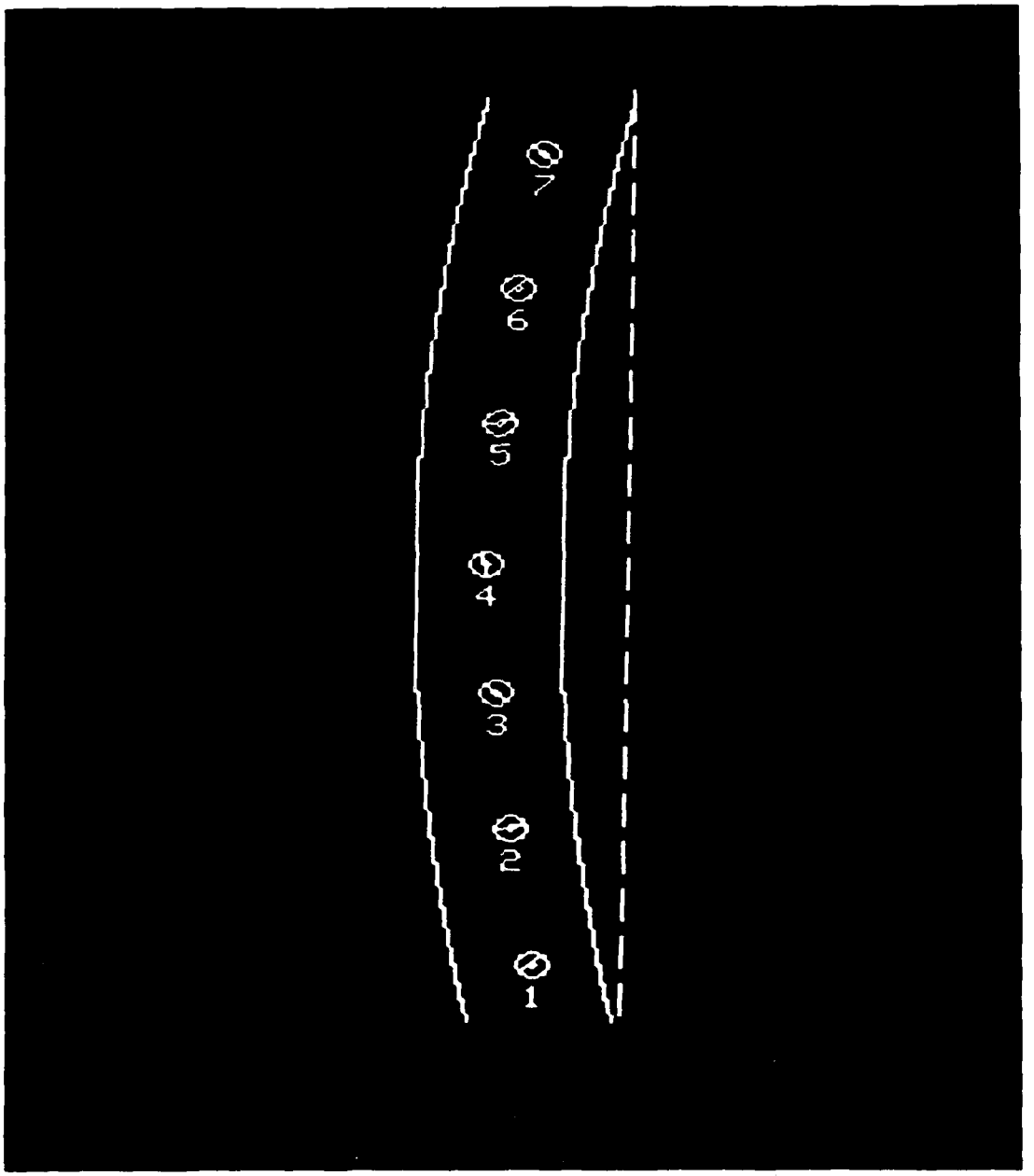


Figure F-4. Fine tuning--graphics display device.

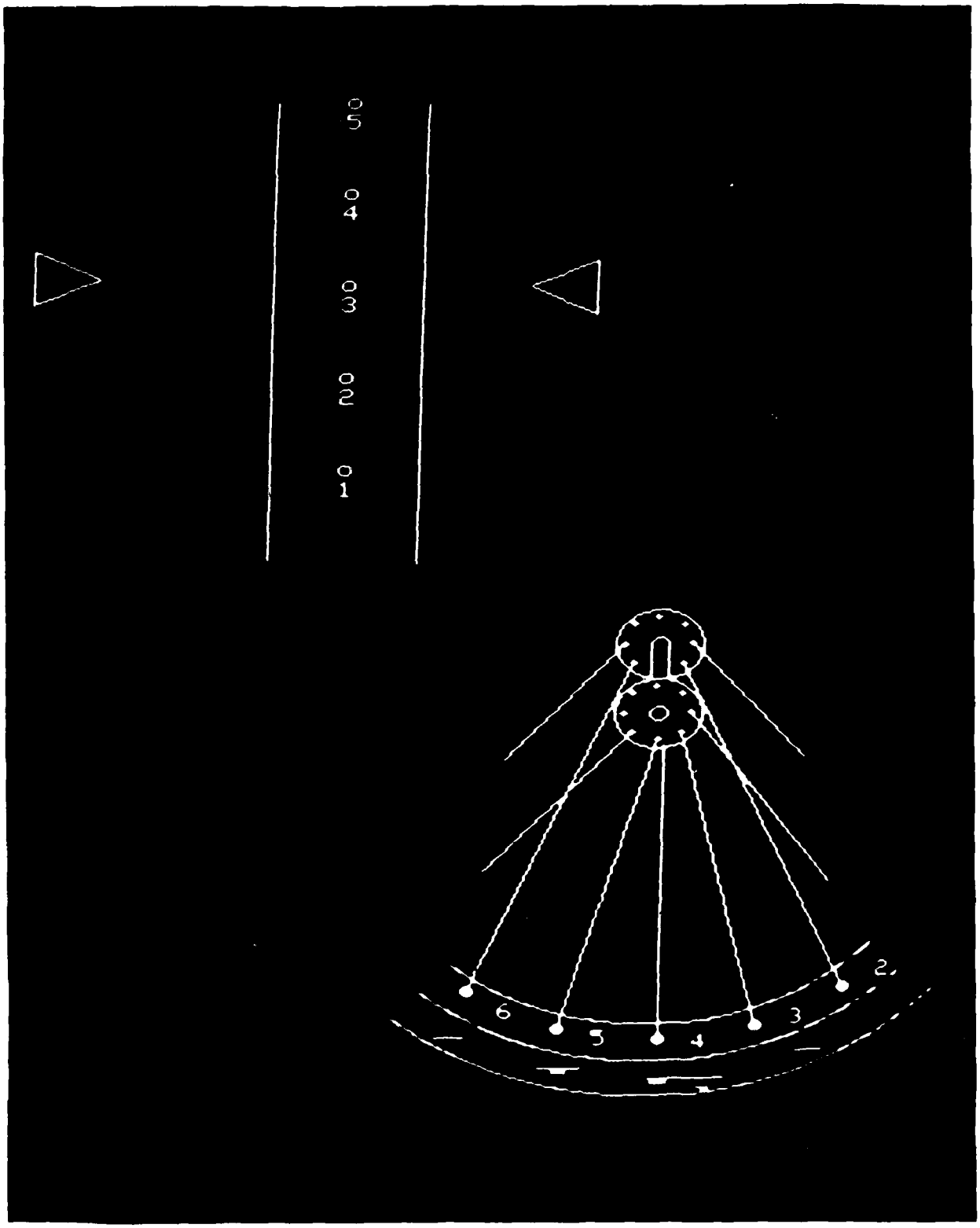


Figure F-5. Graphics driver display for demonstration and practice of wheel truing.

APPENDIX G

ILLUSTRATIONS OF DEVICE LL
(LINE DRAWINGS)

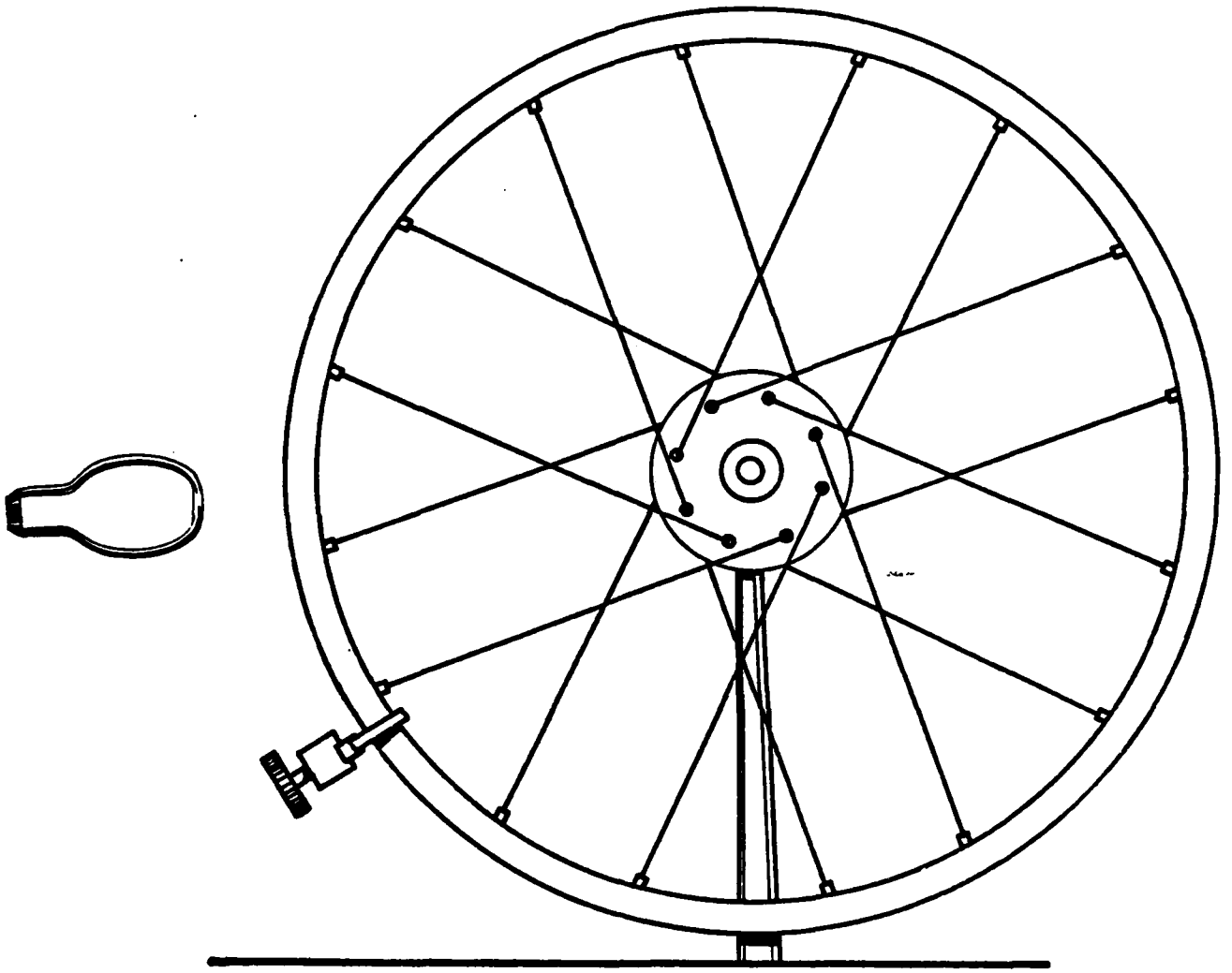


Figure G-1. Equipment components.

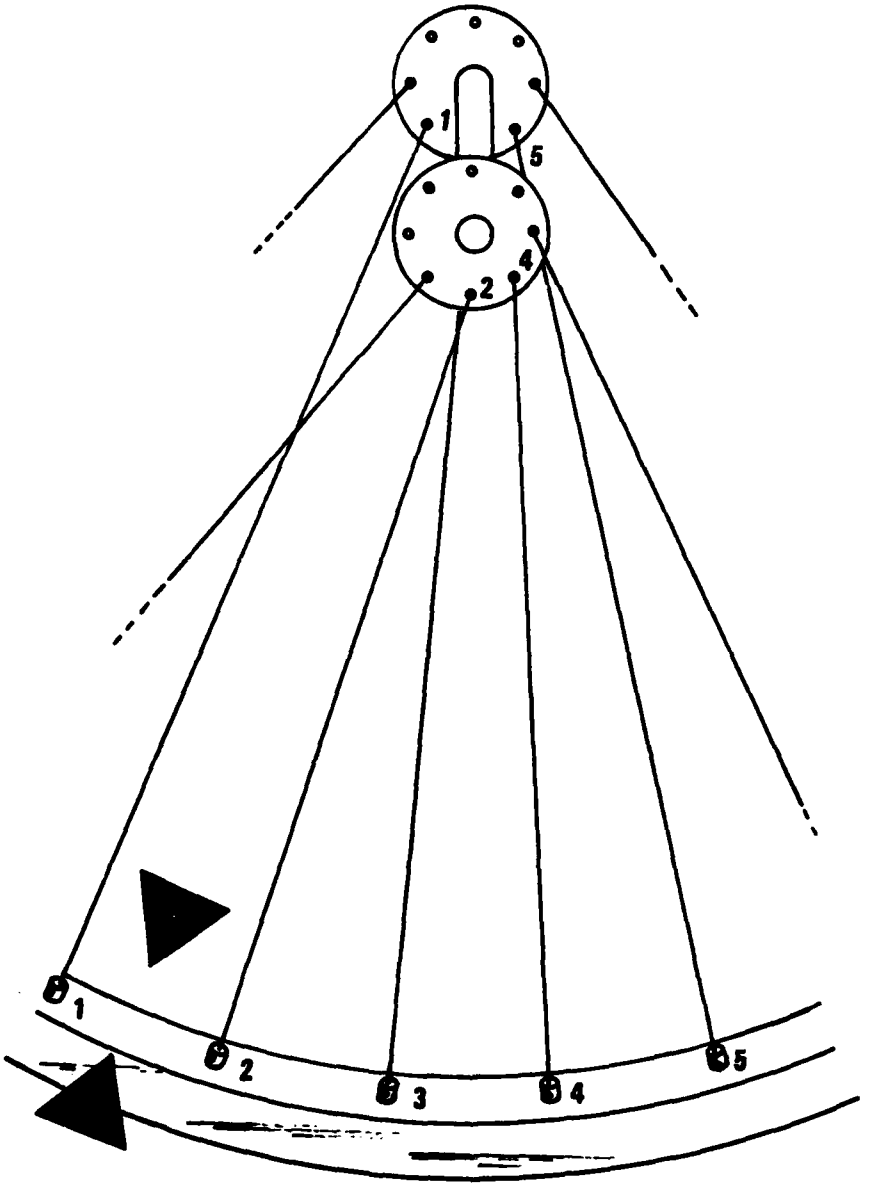


Figure G-2. Finding wobble.

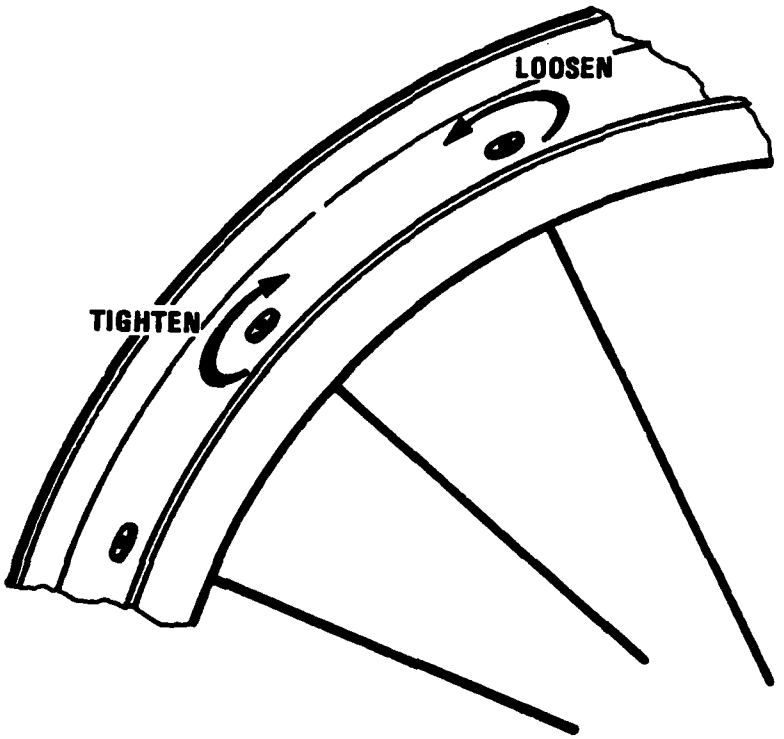


Figure G-3. Spoke adjustment.

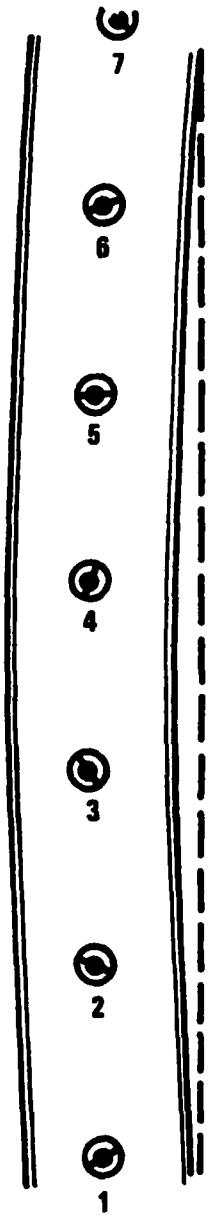


Figure G-4. Fine tuning.

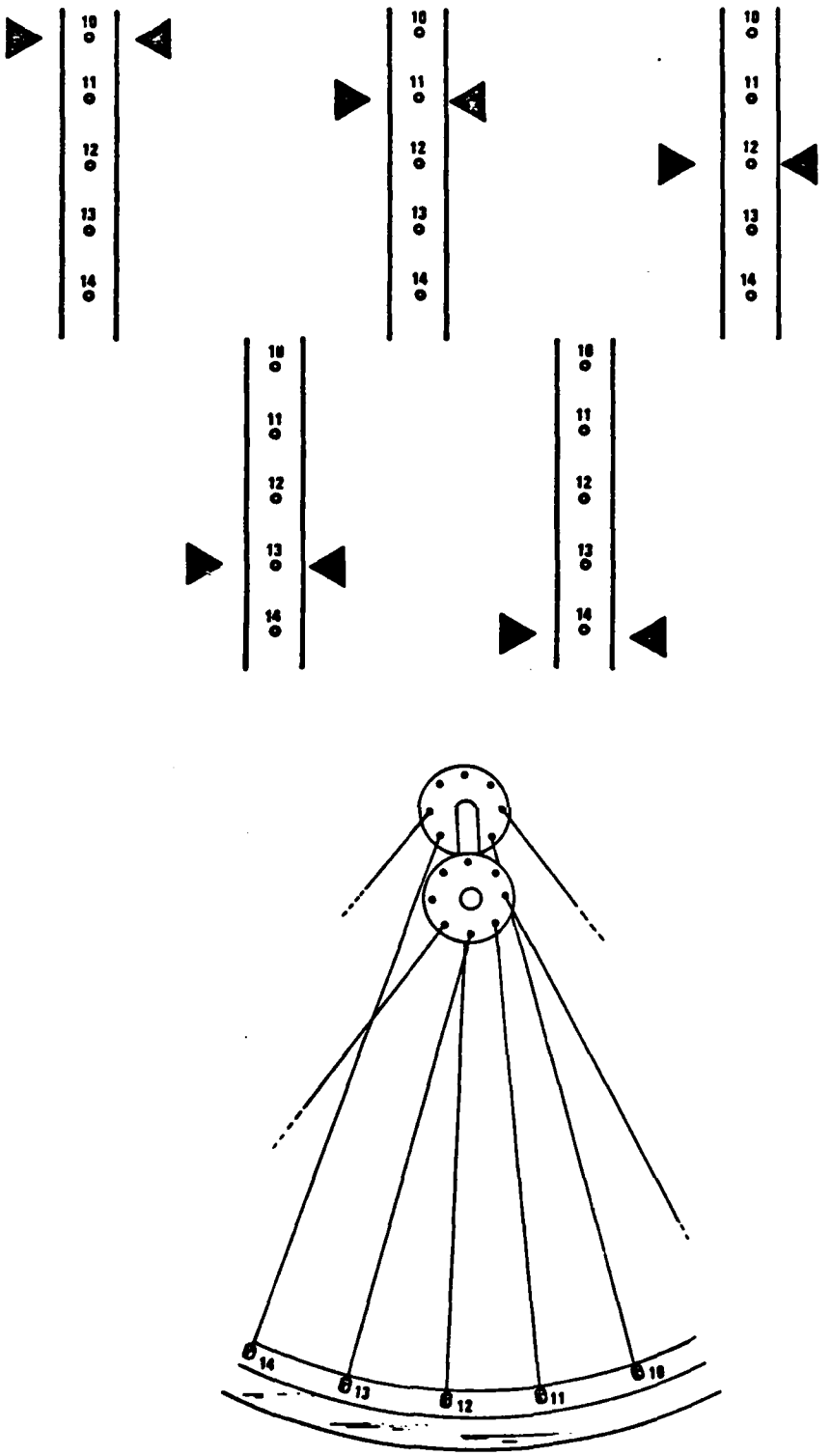
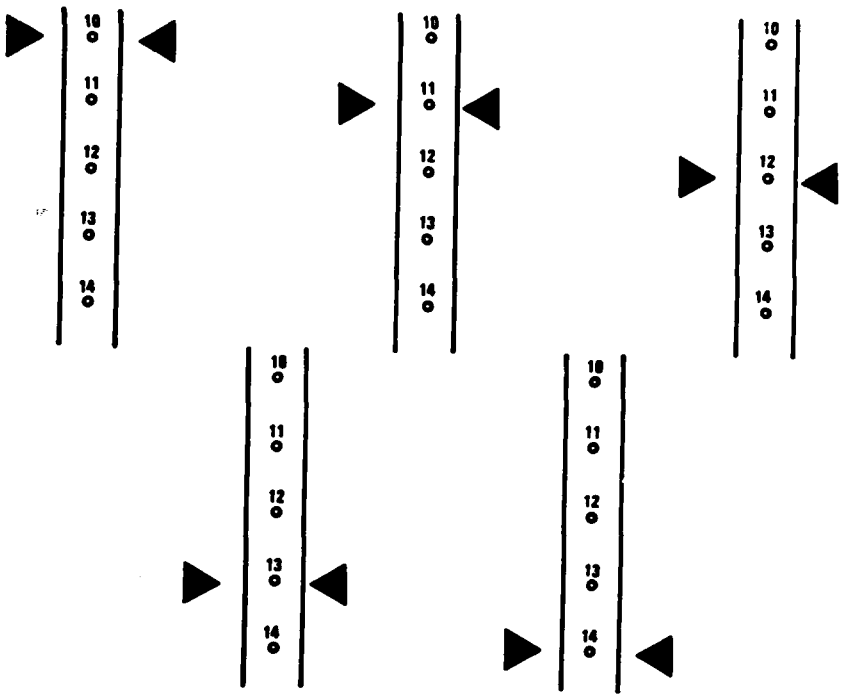


Figure G-5. Simulator exercise (demonstration).



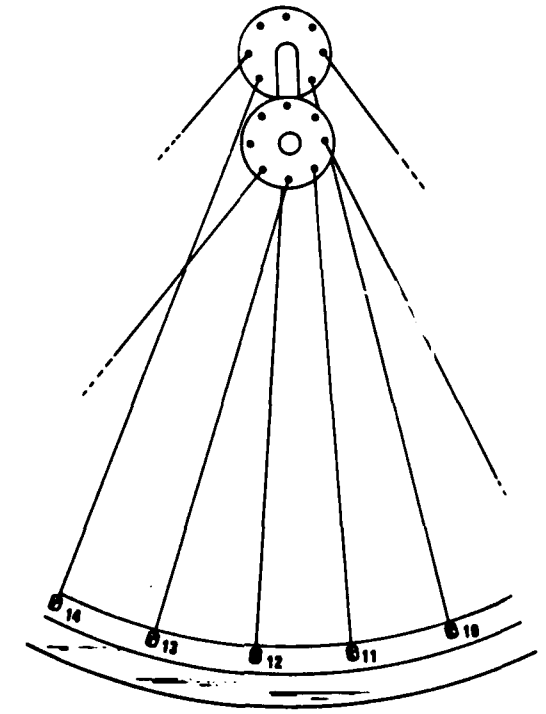


Figure G-6. Exercise 1.

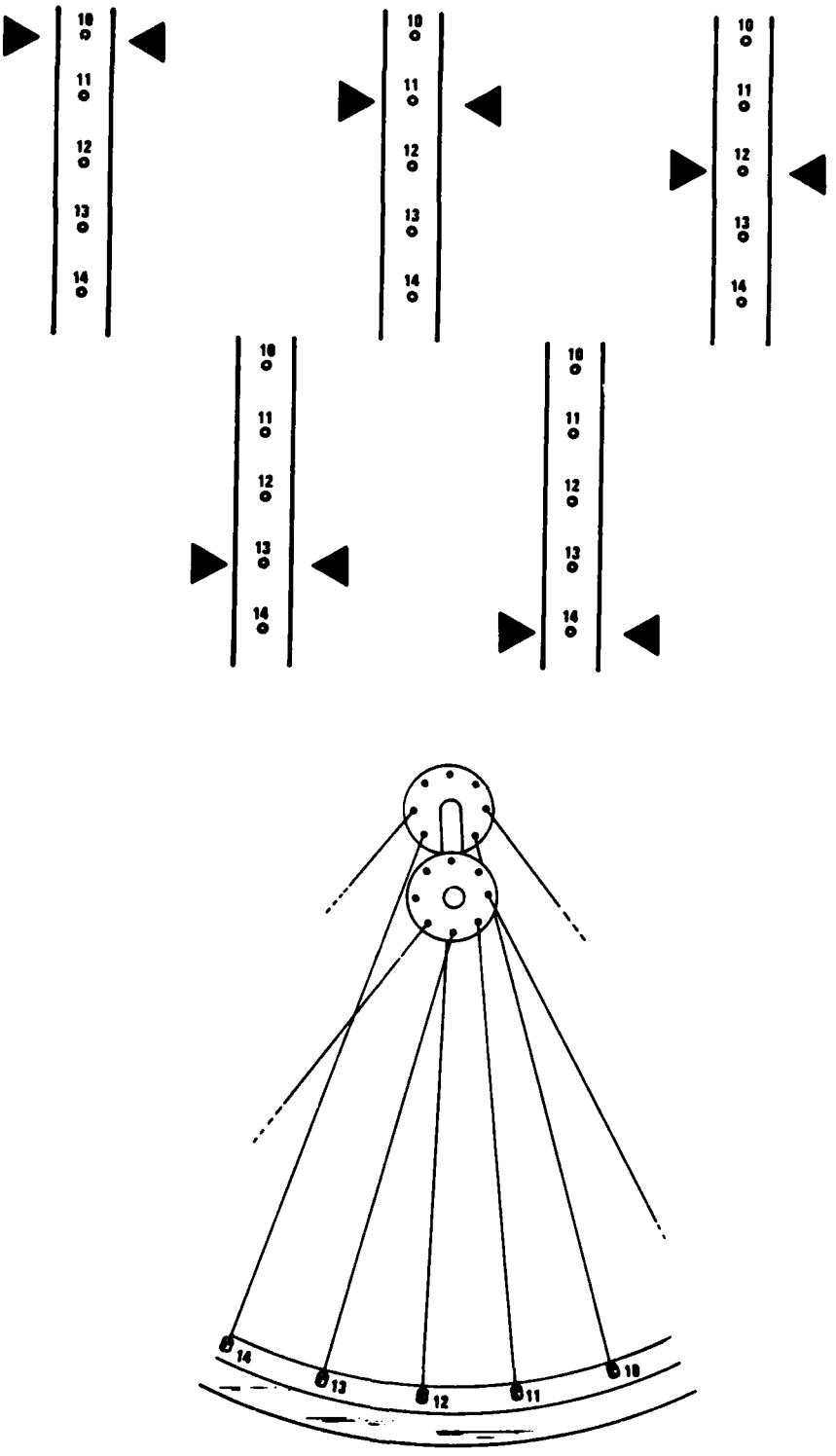
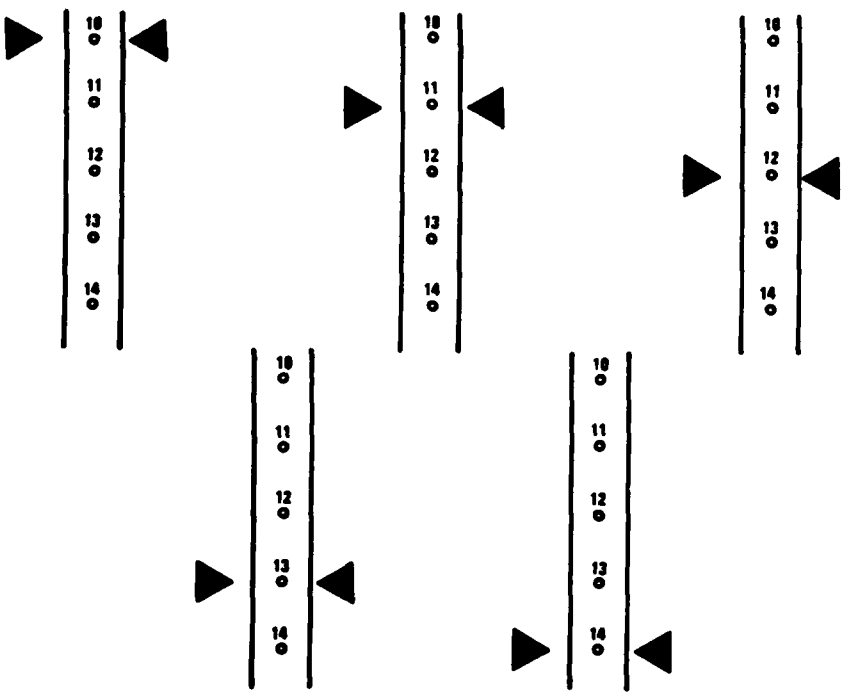


Figure G-7. Exercise 2.



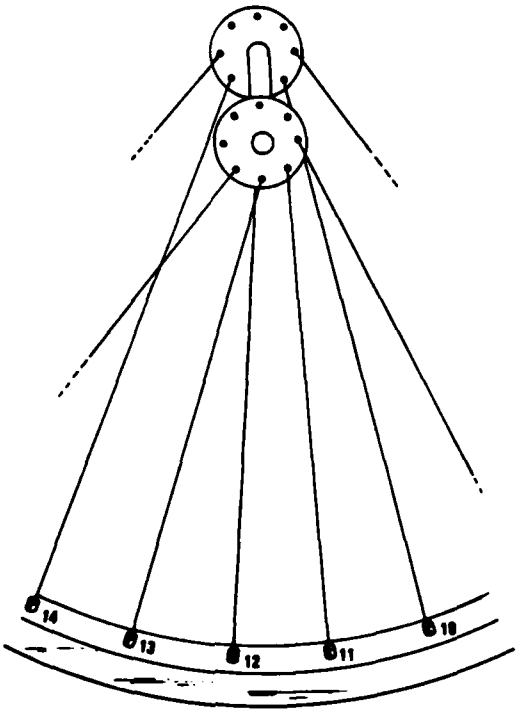


Figure G-8. Exercise 3.

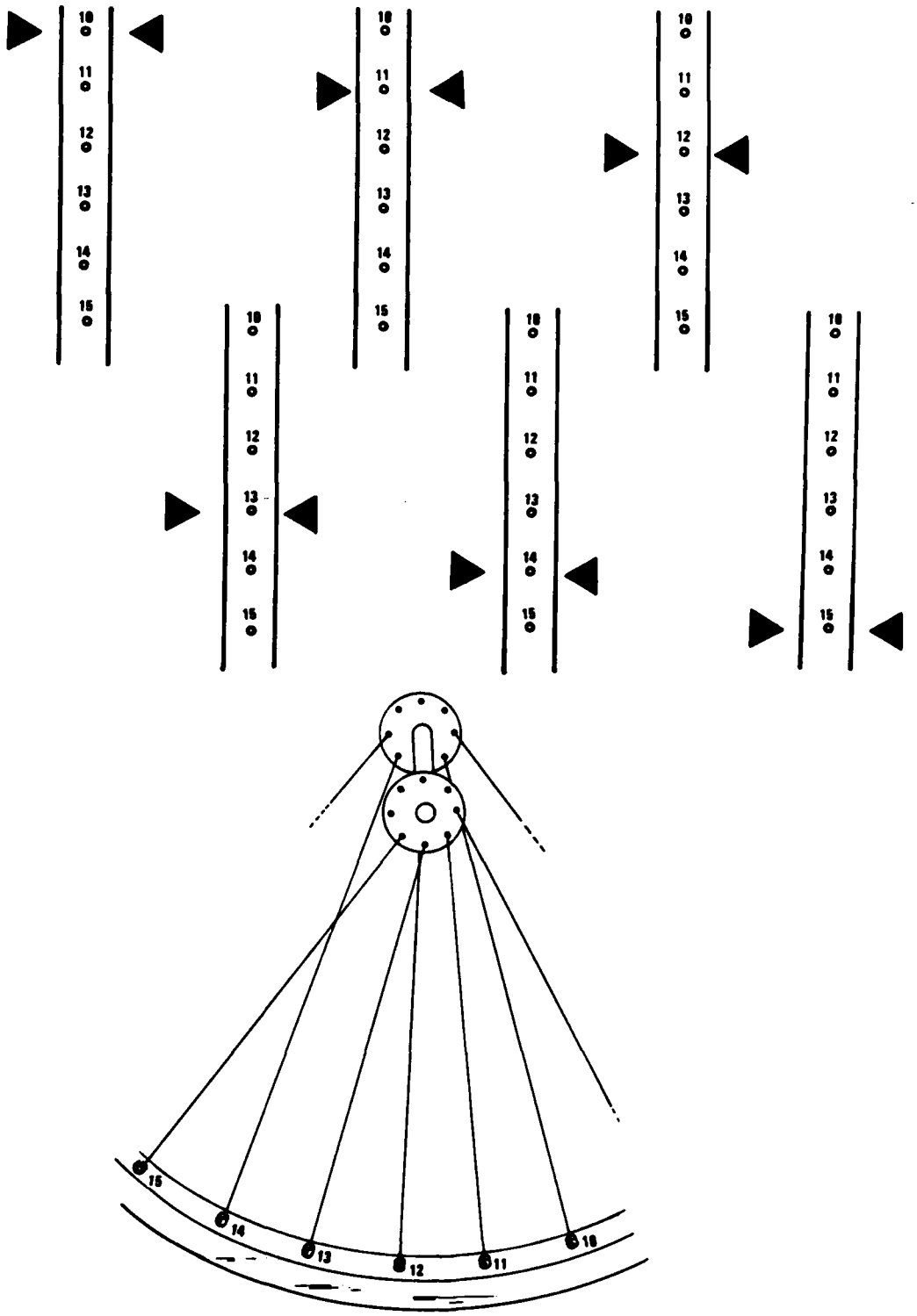


Figure G-9. Exercise 4.

APPENDIX H

INSTRUCTIONS TO SUBJECTS

SIMTRAIN TASK 2 SIMULATOR FIDELITY EXPERIMENT:
SCRIPT AND GENERAL PROCEDURE

I. INTRODUCTION: EXPLANATION OF EXPERIMENT

This study compares different methods of teaching people mechanical skills. We're trying to find out which methods work better. The skill we're studying in this experiment is how to true, or balance, a bicycle wheel--that is, how to get out the side-to-side wobble. First, I'll demonstrate how to true a bicycle wheel using __ (pictures, this copy of a bicycle wheel, computer graphics, or a bicycle wheel, as appropriate). Then you'll have a chance to practice on the __ (pictures ... a bicycle wheel). Finally, I'll put an actual bicycle wheel out of balance and ask you to true it. I'll do this two (or four, for condition HH) times and each time you will have 15 minutes to true it. I'll take a measurement every three minutes. If you have any questions during the demonstration or practice (or first two trials, for condition LL), be sure to ask.

II. DEMONSTRATION BY EXPERIMENTER

Instruction	Action
First, let me demonstrate. The parts we'll be using to true the wheel are: bicycle rim nipples hub spoke wrench spokes caliper	Point out parts on training device. Use Figure G-1 for condition LL). Use Figure F-1 for condition HL.
The caliper id used to find where wheel is out of balance. If the wheel is closer to the left side of the caliper, as it is here, it means the wheel is pulling towards the left at that point. It is off	Point out on device. Use Figure F-2 for condition HL. Use Figure G-2 for condition LL.

Instruction	Action
center here. If it would be closer to the right side, the wheel would be pulling towards the right.	
Once we've found where the wheel is out of balance, how do we fix it? We fix it by loosening or tightening the spokes. If the rim pulls to this side, the reason is that the spoke going to the same side of the hub is too tight; so it needs to be loosened. The spoke going to the other side of the hub should then be tightened so it will pull the rim over toward this side. This means you will need to loosen spokes that go to the same side of the hub that the wheel is pulling towards and tighten spokes that go to the opposite side of the hub.	Point out on device. Use Figure F-2 for HL, and Figure G-2 for LL.
Spokes are tightened and loosened by tightening or loosening these nipples. The nipples are tightened and loosened with the spoke wrench. The spoke wrench fits around the nipples and is used to turn them.	Point out spokes, nipples, spoke wrench. Use Figure F-2 for HL, and Figure G-2 for LL. Demonstrate on LL, MM, and LH.
To tighten a nipple, turn it clockwise as you're looking at it from the outside of the rim. To loosen a nipple, turn it counterclockwise as you're looking at it from the outside.	Demonstrate on HH and LH. Show Figure F-3 for HL, and Figure G-3 for LL. Point out on LH.

Instruction	Action
Wheels usually aren't out of balance only at one spoke; usually a whole line of spokes is out of balance. In this case, the middle spoke needs the most adjustment. It needs to be turned the most, while the spokes at the ends of the group need the least adjusting. For example, if spokes __ to __ (fill in from simulator) are out of alignment, spoke __ (middle spoke) would need perhaps a half-turn while spokes __ __ (end spokes) might need an eighth-turn.	Point out on device. Show Figure F-4 for HL, and Figure G-4 for LL.
(Condition HH)	
The rim here is close to the right caliper, which means it pulls to the right and needs to be pulled back to the left. Spoke __ goes to the right side of the hub so it needs to be loosened. It's loosened by turning it counter-clockwise. The next spoke, __, goes to the left side of the hub. So if we tighten it, the rim will be pulled to the left. It's tightened by turning it clockwise.	Before HH begins, put two deviations in the wheel--one for demonstration and one for practice. True the section of the wheel just pointed out. Continue similarly until the rim section is trued.

Instruction	Action
(Condition MM)	
Turn your head while I put in a deviation. You can look now. The rim here is closer to the right caliper, which means it pulls to the right and needs to be pulled back to the left. The spoke goes to the right side of the hub, so it must be loosened. It's loosened by turning it counterclockwise. The next spoke goes to the left so it needs to be tightened.	Adjust caliper to illustrate a deviation to the right. Demonstrate. Repeat the prior instructions with a second and third deviation, illustrating deviations to the left, and a balanced wheel.
(Condition LL)	
Here these figures represent the rim, spokes, and caliper at different positions on the rim. In 3, the rim is closer to the right side of the caliper. In 2 and 4 it's also closer, and in 1 and 5 the rim looks about midway in the caliper. Spoke 12 is in the middle, so I'll turn it the most, perhaps half a turn. Spokes 11 and 13 I'll turn less, perhaps a quarter-turn, because they are on the ends. Spokes 10 and 14 I won't turn because they are positioned evenly in the caliper. Looking at the bottom picture, spoke 12 goes to the left side of the hub. Since I want to pull the rim toward the left,	Show Figure G-5.

Instruction	Action
I will tighten it by turning it clockwise. Spokes 11 and 13 go to the right side of the hub, and should be loosened in order to pull the rim to the left. These spokes are loosened by turning them counterclockwise.	Make clockwise turning motion.
(Condition LH)	
Turn you head while I adjust the wheel. You can look now. The rim here is closer to the right caliper, which means it pulls to the right and needs to be pulled back. The spoke goes to the right side of the hub, so it must be loosened. It's loosened by turning it counterclockwise. The spoke above it goes to the left, so it needs to be tightened.	Adjust wheel to a position where it is closer to one side of caliper. Make turning motions with wrench. Repeat with two additional positions.
(Condition HL)	
This represents a bicycle rim, these are spokes, and these are caliper. This view of the wheel is included so you can tell which side of the hub the spokes go toward.	Show Figure F-5. Point out parts.
Using the computer, I can do all the things I need to do to true the rim. I can turn the wheel by pressing T and return.	Point out menu on the display. Press T and R (return).

Instruction	Action
I can make the wheel go faster by pressing "W," return, and "Z" for faster and return.	Press W, R, then Z, R.
I can stop the wheel by pressing "S," return.	Press S, R.
I can adjust the caliper by pressing "C," return, and then "I" for in and return.	Press C, R, then I, R.
I can adjust the spokes. For instance, if I want to fix the wheel at the place where it is now, I would press "A," return, to adjust spokes.	Press A, R.
Now the display asks what spoke I want to adjust. I"ll want to adjust "29" so I'll press "29," return.	Press 29, R.
Now I need to tell the computer if I want to loosen or tighten spoke 29. The rim is going to the left, and looking at the bottom picture, spoke 29 is also going to the left side of the hub, so I want to loosen it. To loosen a spoke I should turn it counterclockwise.	Point out on display
The display says counterclockwise is "2" so I will press "2," return.	Press 2, R.

Instruction	Action
This moves the rim a little to the left. It's still not centered in the caliper so the nipple needs to be turned counterclockwise again.	Continue pressing 2, R until rim is centered between caliper.
It's centered now, so I'll stop the adjustments on this spoke by pressing "S," return. I can adjust spokes 30 and 28 similarly. To stop all adjustments on the spokes at this position, I'll press "S," return.	Press S, R.
Now I can turn the wheel again to find another deviation by pressing "T" and return.	Press T, R.
Here's another deviation so I'll stop the wheel here.	Press S, R when the wheel is close to one side of the caliper.
I can fix this deviation just as I did the one before--by pressing "A," then pressing the number of the spoke and turning it clockwise or counter-clockwise. Do you have any questions?	Demonstrate

III. PRACTICE

Now you'll have a chance to practice using __ (pictures, the computer, this copy of a wheel, this wheel).

Instruction	Action
(Condition HH)	
You'll have two 15-minute trials, and I'll take a measurement every three minutes. If you have any questions during these trials, be sure to ask. After these two trials you'll get a second set of two 15-minute trials during which you won't be able to ask questions.	E puts in a standard set of deviations, takes the initial measurement, and calls S back in. S gets one 15-minute trial, after which E trues wheel if necessary and puts in a standard deviation for the second 15-minute trial.
Would you leave the room while I put the wheel out of balance?	Call subject back in.
Go ahead.	
(Condition MM)	
Turn your head, and I'll put in a deviation. Now show me how you would turn the nipples with the spoke wrench. If you have any questions be sure to ask.	E adjusts caliper to a predetermined position. Repeat for three additional deviations.
(Condition LL)	
I'll show you a series of pictures like the one we just went through. I'd like you to tell me which spokes should be adjusted and how much.	Present Figures G-6 thrugh G-10.

Instruction	Action
Also, tell and show me in which direction the nipple should be turned. Be sure to ask questions if there is anything you aren't certain about.	
(Condition HL)	
Would you like to do this now? I'll help you and we'll do it together.	E puts the second exercise on the screen. S sits in front of the CRT and works for 15 minutes. E takes a measurement every three minutes.
(Condition LH)	
Turn your head and I'll adjust the wheel to show a deviation.	Adjust wheel so wheel is closer to one side of caliper.
Now show me how you would turn this nipple with the spoke wrench to fix the deviation. If you have any questions be sure to ask.	Repeat for three additional deviations.

IV TWO PERFORMANCE TRIALS (ALL CONDITIONS)

Instruction	Action
	For each trial
Now if you'll leave the room for a few minutes, I'll put the wheel out of true.	(a) E trues wheel (b) E puts in predetermined amount of deviation into rim (c) E measures initial deviation
(Call subject back in.)	(d) S instructed to true wheel

Instruction	Action
Now see if you can true the wheel. You'll have 15 minutes and I'll take a measurement every three minutes	1. E takes deviation measurements every three minutes 2. Trial continued for 15 minutes, excluding measurement time (e) E does not answer questions

V. POST-EXPERIMENT PROCEDURES

Thank subject. Give him/her $15, ask subject to sign a form acknowledging receipt of the money, and answer any questions the subject may have.

108

APPENDIX I

RAW DATA

APPENDIX I

RIM DEVIATIONS* BY SUBJECT AND CONDITION

Condition	Subject No.	Practice 1 Measurements						Performance 1 Measurements						Performance 2 Measurements					
		Initial	1	2	3	4	5	Initial	1	2	3	4	5	Initial	1	2	3	4	5
HN	1	.2151	.1227	.0578	.0473	.0340	.0340	.2213	.1068	.0646	.0553	.0536	.0513	.1536	.0904	.1037	.0819	.0816	.0771
	2	.1400	.0669	.0618	.0434	.0385	.0388	.2114	.0674	.0521	.0434	.0499	.0530	.1836	.0544	.0581	.0567	.0592	.0553
	3	.1782	.1743	.2139	.1896	.1179	.1131	.2420	.2309	.2187	.1966	.1731	.1510	.1859	.1473	.0776	.0697	.0510	.0417
	4	.2278	.3023	.2261	.1986	.4137	.6194	.2777	.0612	.0632	.4652	.1816	.0793	.2644	.0638	.0853	.0468	.0519	.0507
	5	.2474	.2383	.2111	.0847	.0567	.0453	.3247	.1218	.0901	.0751	.0669	.0714	.2763	.0612	.0300	.0468	.0448	.0490
	6	.2425	.3156	.3097	.2556	.2488	.2343	.2822	.2403	.2638	.2683	.2063	.1411	.2335	.3375	.2700	.2882	.3060	.2142
	7	.2607	.4913	.2080	.1485	.1250	.0901	.2853	.1898	.1145	.1040	.1111	.1431	.3171	.3601	.1145	.0618	.0961	.0980
	8	.2326	.0604	.0870	.0706	.1482	.0757	.3352	.0870	.0638	.0411	.0391	.0334	.2559	.3556	.0507	.0309	.0334	.0366
	9	.2357	.2326	.3275	.2125	.2244	.2567	.1420	.1692	.0978	.1193	.0714	.0677	.2901	.2638	.1938	.2017	.1301	.1148
	10	.2522	.2391	.1564	.2366	.1966	.1278	.2714	.3678	.5279	.3018	.3791	.3134	.2134	.5687	.4293	.3284	.3069	.5998
	11	.2910	.2675	.1278	.0720	.0717	.0776	.2802	.1102	.0300	.0402	.0337	.0360	.3494	.0859	.1063	.0708	.0272	.0272
	12	.3111	.0912	.1187	.1459	.1034	.1102	.2394	.3958	.2567	.2428	.1448	.0938	.2669	.1896	.1312	.0734	.0601	.0436
	13	.2241	.2298	.2513	.2593	.0745	.0598	.3179	.3723	.2663	.1278	.1074	.0708	.2236	.1380	.1235	.0612	.0303	.0385
	14	.2669	.2284	.1587	.0649	.0261	.0232	.3202	.2083	.1918	.1669	.0589	.0448	.2448	.0808	.0289	.0315	.0167	.0082
	15	.2632	.0833	.0295	.0261	.0244	.0193	.3222	.0623	.0541	.0303	.0264	.0139	.2777	.0405	.0255	.0170	.0258	.0159
	16	.2550	.2170	.1383	.1269	.1145	.1298	.2873	.1785	.1125	.0839	.0995	.2003	.2493	.2139	.0992	.0674	.0470	.0422
	17	.2431	.0431	.0507	.0295	.0414	.0340	.2913	.0635	.0422	.0584	.0601	.0422	.2527	.0320	.0193	.0278	.0247	.0238
	18	.2754	.2159	.0833	.0694	.0366	.0615	.2896	.1193	.0589	.1516	.0400	.0213	.2686	.0975	.0201	.0068	.0031	.0065
	19	.2598	.1629	.1071	.0737	.0397	.0042	.2986	.1068	.0394	.0315	.0176	.0181	.2822	.1998	.0632	.0283	.0340	.0167
	20	.2714	.3658	.2765	.3426	.2454	.2165	.2850	.1119	.0609	.0462	.0357	.0621	.2595	.1927	.0312	.0332	.0349	.0170
HL	1	.2260	.1735	.1806	.1524	.0924	.0910	.2714	.2525	.2284	.1828	.0842	.0612	.3610	.2380	.1403	.0740	.0697	.0819
	2	.2260	.2271	.2375	.2375	.2462	.2424	.2284	.3791	.2278	.3154	.4358	.3295	.2771	.2051	.2221	.1969	.3366	.4488
	3	.2260	.1839	.1689	.1689	.1716	.1716	.2774	.2916	.2610	.2675	.2828	.2802	.3026	.2661	.2763	.3171	.3499	.3635
	4	.2260	.1715	.1650	.1672	.1550	.1533	.2748	.2408	.1193	.1743	.0697	.2570	.1932	.0816	.0595	.0487	.0587	.0570
	5	.2260	.1718	.1553	.1744	.1704	.1610	.2536	.2505	.2652	.3709	.2542	.2510						
	6	.2260	.1627	.1559	.1473	.1258	.1138	.2457	.3173	.1085	.1873	.1757	.1417	.2346	.0572	.0380	.0238	.0465	.0513
	7	.2260	.1709	.1706	.1406	.1053	.1003	.2292	.2448	.1354	.1295	.0850	.0827	.2870	.1193	.0672	.0609	.0547	.0363
	8	.2260	.1872	.1696	.1245	.0985	.0859	.2397	.2414	.1091	.1014	.0754	.0649	.2845	.2151	.1890	.0720	.0527	.0504
	9	.2260	.1442	.1082	.0538	.0537	.0282	.2746	.0411	.0278	.0400	.0213	.0159	.2516	.0655	.0354	.0241	.0153	.0150
	10	.2260	.2100	.2100	.1772	.1772	.1190	.2638	.1794	.0887	.0349	.0221	.0062	.2831	.0966	.0487	.0380	.0303	.0306
	11	.2260	.2072	.2469	.1524	.1251	.1202	.2527	.2434	.1544	.0439	.0340	.0232	.2941	.0768	.1000	.0439	.0147	.0258
	12	.2260	.2003	.1734	.1412	.1189	.0847	.2638	.2553	.2049	.0567	.0462	.0303	.2683	.1136	.0337	.0201	.0269	.0241
	13	.2260	.2054	.1641	.1603	.1391	.1322	.2641	.1046	.0337	.0368	.0249	.0136	.3372	.0502	.0436	.0309	.0210	.0258
	14	.2260	.1655	.1655	.1397	.1201	.0982	.2612	.2230	.2210	.2179	.1686	.1179	.3312	.3352	.4950	.5633	.5905	.6497
	15	.2260	.2393	.2076	.2278	.1958	.1476	.2448	.1544	.1374	.1196	.1910	.1383	.2961	.1995	.1451	.1204	.0944	.0595
	16	.2260	.2195	.2605	.2603	.2601	.2601	.2397	.2187	.2771	.1983	.1989	.1425	.2782	.1896	.1233	.0842	.1034	.0533
	17	.2260	.2224	.1966	.1997	.2072	.2053	.2573	.1272	.1199	.1686	.1680	.0910	.3026	.2168	.1150	.0592	.0434	.0340
	18	.2260	.2124	.1659	.1316	.1484	.1447	.2516	.3536	.2060	.1573	.1431	.1173	.2856	.1969	.1411	.1014	.0706	.0626
	19	.2260	.3397	.2651	.2852	.3003	.2541	.2397	.2584	.2298	.2697	.2301	.2003	.3046	.2502	.2502	.2646	.2731	.2712
	20	.2260	.2335	.2335	.2374	.2374	.2295	.2544	.1964	.2400	.1828	.1714	.1340	.3029	.1983	.1380	.1295	.0629	.0782

*Measured in inches

APPENDIX I

RIM DEVIATIONS* BY SUBJECT AND CONDITION

Condition	Subject No.	Practice 1 Measurements						Performance 1 Measurements						Performance 2 Measurements					
		Initial	1	2	3	4	5	Initial	1	2	3	4	5	Initial	1	2	3	4	5
MM	1							.1482	.1448	.1439	.1326	.1167	.1068	.2247	.2088	.1652	.1099	.0935	.0912
	2							.1839	.1734	.1765	.1842	.1677	.1598	.2570	.2763	.2238	.2391	.2267	.0606
	3							.1609	.1924	.1301	.0893	.0332	.0431	.2595	.1326	.0901	.0289	.0346	.0320
	4							.2049	.1672	.0918	.0567	.0476	.0289	.3573	.1734	.0986	.0490	.0306	.0340
	5							.2006	.1533	.0592	.0626	.0550	.0317	.3318	.2012	.0822	.0759	.0638	.0547
	6							.2652	.2564	.0666	.0527	.0584	.0377	.2921	.1286	.0898	.0589	.0612	.0536
	7							.2510	.2995	.2731	.2845	.2332	.2054	.3122	.1697	.1496	.1204	.0774	.0734
	8							.2794	.2944	.4120	.3791	.4851	.6797	.2780	.2819	.2539	.2083	.1856	.1465
	9							.2655	.2601	.2561	.2295	.1726	.1564	.2729	.2816	.2879	.2496	.2726	.2372
	10							.2774	.2046	.2289	.1969	.1471	.1408	.3202	.3190	.2502	.2003	.1791	.1468
	11							.2117	.1913	.0740	.0561	.0445	.0575	.3012	.2754	.2287	.1751	.1598	.1482
	12							.2261	.3176	.2386	.1791	.1213	.0850	.2961	.1989	.2335	.2142	.2185	.1703
	13							.2338	.3236	.6344	.4675	.6276	.2459	.2692	.3139	.2850	.3066	.4094	.5432
	14							.2587	.2570	.2454	.2185	.0898	.0397	.2831	.1360	.0669	.0366	.0292	.0315
	15							.2236	.2034	.1587	.0584	.0439	.3074	.3100	.2074	.1252	.0714	.0638	.0374
	16							.2621	.2315	.0748	.0618	.0558	.0434	.2527	.0997	.0629	.0796	.0544	.0822
	17							.2465	.1700	.1686	.1564	.1235	.0946	.2505	.0850	.0802	.0326	.0397	.0315
	18							.2468	.1544	.1193	.0771	.0507	.0606	.3202	.1629	.0825	.0674	.0224	.0156
	19							.2652	.1723	.0431	.0422	.0130	.0264	.3400	.1363	.1247	.1125	.0649	.0615
	20							.2590	.3774	.5145	.2015	.2910	.0853	.2816	.2088	.2355	.0553	.0473	.0555
LH	1							.2100	.2080	.2176	.1065	.0615	.0697	.2610	.3375	.2026	.1590	.1740	.1099
	2							.1652	.1371	.1060	.1048	.0938	.0963	.2428	.2323	.1904	.1652	.1046	.0980
	3							.2380	.1893	.0776	.0524	.0425	.0258	.3202	.1774	.0793	.0385	.0329	.0332
	4							.2593	.3241	.2216	.2326	.1966	.1998	.3117	.4097	.2896	.2828	.2499	.1760
	5							.2440	.1989	.2102	.1221	.0944	.0436	.2890	.1082	.0635	.1000	.0448	.0932
	6							.2626	.2049	.2264	.2901	.2287	.1898	.2930	.2054	.0884	.0510	.0635	.0490
	7							.2380	.1887	.1207	.0663	.0487	.0448	.3230	.1391	.0564	.0201	.0156	.0213
	8							.2828	.3091	.2332	.1949	.1830	.2006	.2760	.3049	.2646	.2649	.2204	.1598
	9							.2434	.2219	.1394	.1227	.1437	.0844	.3386	.2904	.1652	.1879	.1717	.0785
	10							.2513	.3315	.1703	.1761	.0839	.1085	.3001	.1791	.0938	.1340	.0907	.0819
	11							.2264	.3154	.0765	.0720	.0490	.0142	.3389	.2414	.0796	.0598	.0377	.0249
	12							.2468	.1726	.0910	.0740	.0618	.0669	.2655	.1235	.1408	.0371	.0666	.0536
	13							.2539	.2845	.3029	.3205	.2224	.2508	.2848	.1635	.0567	.0150	.0162	.0142
	14							.2525	.1683	.1292	.0618	.0224	.0179	.3224	.1530	.0649	.0329	.0388	.0408
	15							.2890	.0873	.0751	.1048	.0929	.1006	.2921	.1340	.1080	.0643	.0649	.0677
	16							.2425	.0949	.0499	.0281	.0380	.0286	.3188	.1700	.0374	.0306	.0210	.0176
	17							.2593	.1833	.0445	.0289	.0502	.0459	.2981	.0439	.0558	.0388	.0802	.0765
	18							.2578	.2723	.2488	.2884	.2227	.1813	.2950	.1788	.1825	.1408	.1921	.1677
	19							.2624	.3003	.2488	.0400	.0544	.0541	.3233	.0822	.0337	.0482	.0445	.0445
	20							.2550	.1578	.0762	.0689	.0374	.0587	.2867	.0485	.0196	.0167	.0261	.0179

*Measured in inches

APPENDIX I

RIM DEVIATIONS* BY SUBJECT AND CONDITION

Condition	Subject No.	Practice 1 Measurements						Performance 1 Measurements						Performance 2 Measurements					
		Initial	1	2	3	4	5	Initial	1	2	3	4	5	Initial	1	2	3	4	5
LL	1							.1862	.1881	.2029	.2009	.2865	.3360	.2411	.2255	.1686	.0935	.0694	.0414
	2							.1896	.2933	.2505	.2678	.2802	.2522	.2916	.1621	.2071	.0912	.0536	.0615
	3							.2635	.2352	.1587	.1048	.0827	.0686	.2190	.2692	.1550	.1159	.0788	.0643
	4							.1972	.2567	.1046	.0408	.0587	.0442	.3120	.1272	.0955	.0567	.0312	.0162
	5							.2040	.1624	.1442	.1533	.3020	.0935	.2584	.1700	.1125	.0632	.0238	.0255
	6							.1972	.2091	.1261	.0961	.0425	.0425	.2576	.1754	.0652	.0312	.0323	.0343
	7							.2468	.3754	.2349	.3519	.2754	.2833	.2805	.2995	.3836	.1782	.1417	.1065
	8							.2542	.2587	.3049	.2632	.2400	.2627	.3003	.2474	.1731	.1581	.1454	.0912
	9							.2539	.2570	.2559	.2998	.2969	.3165	.3159	.3156	.3122	.3097	.4261	.4964
	10							.2763	.2471	.2476	.2508	.2533	.2559	.2468	.2791	.2536	.3411	.2527	.2533
	11							.2663	.2284	.1706	.1536	.1374	.1196	.3292	.2457	.2383	.2403	.2284	.3134
	12							.2221	.2216	.1040	.0482	.0516	.0074	.2474	.1794	.0414	.0969	.0173	.0227
	13							.2627	.3375	.3961	.4301	.2108	.1703	.3258	.3975	.2513	.1366	.1167	.1598
	14							.2363	.3610	.2632	.2765	.2561	.2559	.2697	.3601	.1768	.2298	.1510	.1363
	15							.2389	.0516	.0300	.0295	.0266	.0264	.2981	.0632	.0201	.0181	.0278	.0275
	16							.2431	.2267	.1128	.0405	.0258	.0289	.3001	.1340	.0530	.0283	.0394	.0247
	17							.2482	.3768	.5060	.4516	.4086	.2879	.3550	.1867	.1522	.1405	.1267	.0683
	18							.2627	.2870	.0561	.0400	.2066	.0360	.3593	.1689	.0553	.0385	.0326	.0340
	19							.2264	.1360	.1250	.1890	.1116	.0504	.2822	.0431	.0499	.0479	.0261	.0346
	20							.2539	.1864	.1879	.1791	.1601	.1567	.2780	.2238	.1862	.1386	.1193	.1046

APPENDIX J

MEANS AND STANDARD DEVIATIONS OF RAW DATA BY MEASUREMENT, TRIAL, AND CONDITION

	Trial	Measurement	Condition HH (N = 20)	HL (N = 20*)	MM (N = 20)	LH (N = 20)	LL (N = 20)	Experts (A = 3)
Practice								
	1	Initial	0.24467	0.22600				
	1	1	0.20742	0.20240				
	1	2	0.16006	0.19004				
	1	3	0.13487	0.17396				
	1	4	0.11906	0.16243				
	1	5	0.11856	0.14716				
	1	Initial	0.27624	0.25440	0.23352	0.24702	0.23646	0.25897
	1	1	0.16855	0.22867	0.22723	0.21751	0.24479	0.07773
	1	2	0.13347	0.16976	0.20548	0.15329	0.19909	0.03797
	1	3	0.13248	0.16277	0.15932	0.12529	0.19337	0.01936
	1	4	0.09780	0.14411	0.14888	0.10139	0.17668	0.02446
	1	5	0.08540	0.12494	0.11831	0.09411	0.15473	0.01813
Performance	2	Initial	0.25241	0.28819	0.29051	0.29904	0.28840	0.30582
	2	1	0.17866	0.16692	0.19988	0.18614	0.21367	0.16840
	2	2	0.10307	0.14008	0.16081	0.11363	0.15754	0.05006
	2	3	0.08150	0.11963	0.12459	0.09438	0.12772	0.02201
	2	4	0.07323	0.12185	0.11671	0.08779	0.10700	0.02522
	2	5	0.07884	0.12731	0.10533	0.07130	0.10581	0.02937
Practice	1	Initial	0.03763	0.00000				
	1	1	0.11269	0.04186				
	1	2	0.08912	0.04096				
	1	3	0.09162	0.05400				
	1	4	0.10168	0.06324				
	1	5	0.13808	0.06353				
Performance								
	1	Initial	0.04626	0.01470	0.03795	0.02596	0.02807	0.00858
	1	1	0.10604	0.08062	0.06748	0.07489	0.08153	0.08146
	1	2	0.12272	0.07596	0.15697	0.07909	0.11442	0.01747
	1	3	0.11395	0.09579	0.11682	0.09090	0.12952	0.01356
	1	4	0.08621	0.10641	0.15823	0.07068	0.11790	0.00548
	1	5	0.07324	0.09576	0.14638	0.07139	0.11502	0.01076
	2	Initial	0.04559	0.03686	0.03393	0.02584	0.03777	0.01944
	2	1	0.13937	0.08208	0.07120	0.09470	0.09128	0.16186
	2	2	0.09924	0.11315	0.07997	0.07777	0.09826	0.03204
	2	3	0.08746	0.13529	0.08468	0.08194	0.09288	0.00976
	2	4	0.08482	0.15386	0.10304	0.07306	0.10192	0.01597
	2	5	0.13151	0.17612	0.11832	0.05055	0.12144	0.01286

*N = 19 for Performance Trial 2.